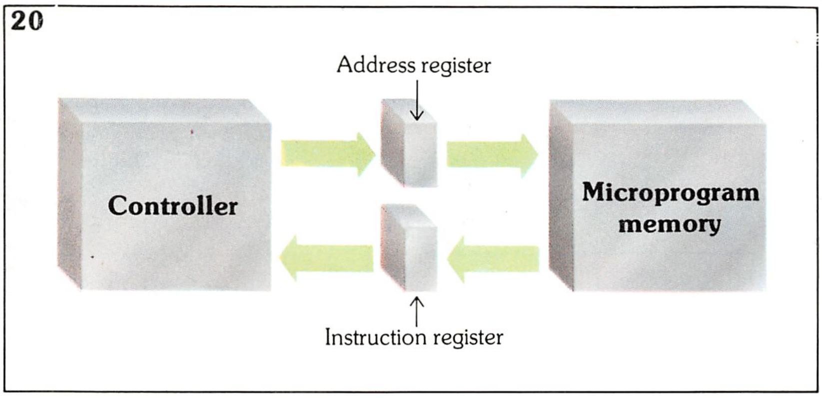
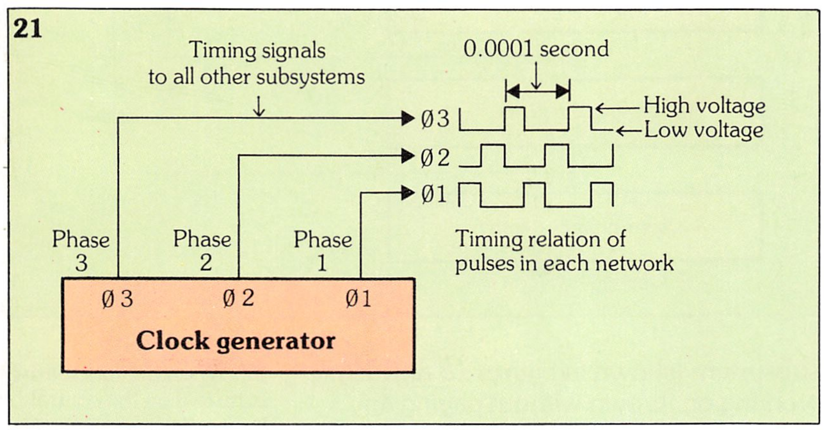
*continued from part 1*

Each of these paths is in fact made up of a number of connection wires or lines that run next to each other carrying electrical signals. To give some idea of the simplification it may help to know that some of these blocks represent thousands of transistors, and a single path may consist of 10 or more individual lines.

**20. The controller** stores the desired instruction address in the address register. The microprogram memory then stores the instruction in the instruction register.




**21. The clock generator** produces sequential pulses in three networks going to all other subsystems, to synchronize their operations.

## How the subsystems are controlled

The first thing to understand is that all subsystems are directly linked to the controller by means of a network of electrical conductors, that are not fully illustrated in *figure 19*. The job of the controller is to tell each subsystem when to act and what to do. The controller itself acts merely as an interpreter of the instructions which it receives, one at a time, from the place in which they were memorized when the chip was constructed. This place is called a **microprogram memory.**

*Figure 20* shows the process. Each instruction is permanently memorized in a particular location of the microprogram memory. Each memory location, and therefore the instruction which it contains, is identified by a number called an address, rather like the number of your house. The controller receives in succession each instruction which it needs by lodging the relevant address in a temporary memory unit. The unit is called the **address register.**

The microprogram memory goes immediately to the address indicated, where it traces the instruction and transfers a copy of it to another temporary memory unit called the **instruction register.** From here it is read by the controller. As you can see, a register is a storage unit which can be used to hold information for a short time, while it is in use.) Each instruction the controller receives in this way is carried out in a period of time called an instruction cycle.

An **instruction cycle** corresponds to the time in which a scan line is energized – about 100 microseconds (100 millionths of a second).

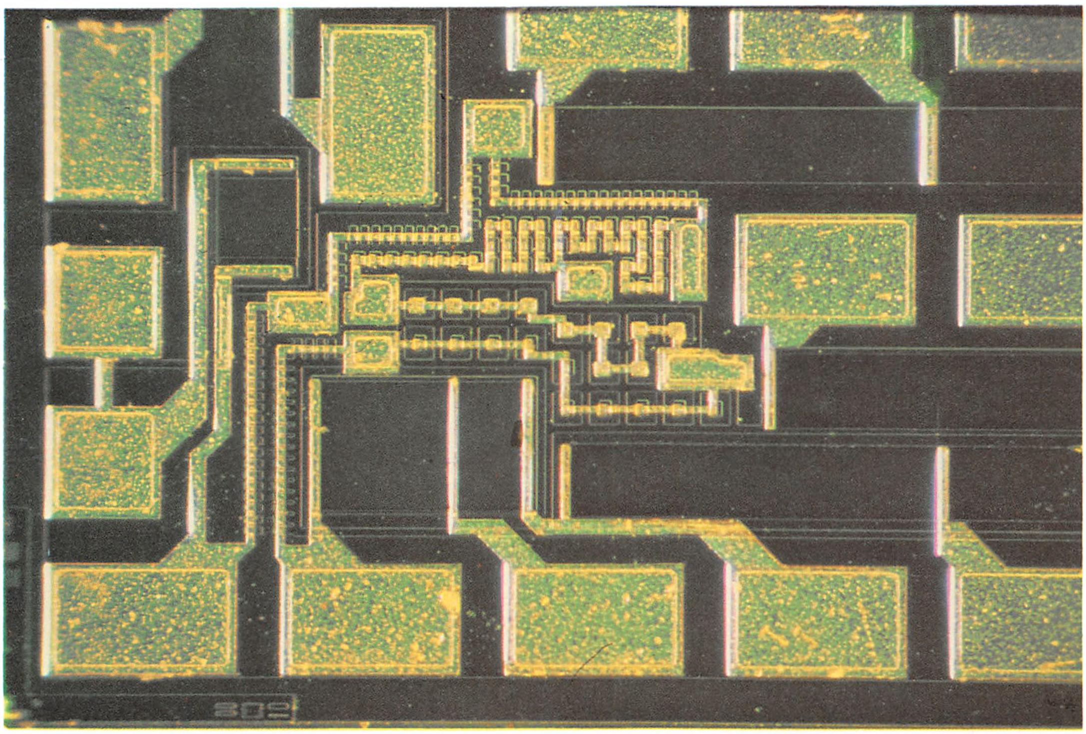
At the end of each instruction cycle, the controller fetches another instruction for the next cycle – stepping along one instruction at a time every cycle.

If the current instruction doesn't tell the controller which instruction to use next, the controller automatically picks the one at the next address in sequence that is lodged in the microprogram memory of the computer.

## How are operations synchronized?

Obviously, timing is a very important aspect of the calculator's work. The operation of all subsystems is 'synchronized' (kept in step) by timing pulses in three different wiring networks, which are illustrated in *figure 21*.

These pulses, called clock signals, are supplied to all parts of the IC from a main timing subsystem called the **clock generator**. The three networks, and the pulses each one carries, are called 'phase one, phase two and phase three'. The three pulses occur one after the other in a regular rhythm, like someone calling out waltz time: 'One-two-three, one-two-three'. Certain parts of the system will not go into action until they receive these phased timing signals.



Science Photo Library

**Left: A 64K PROM** (Programmable Read Only Memory) microchip magnified 22 times.

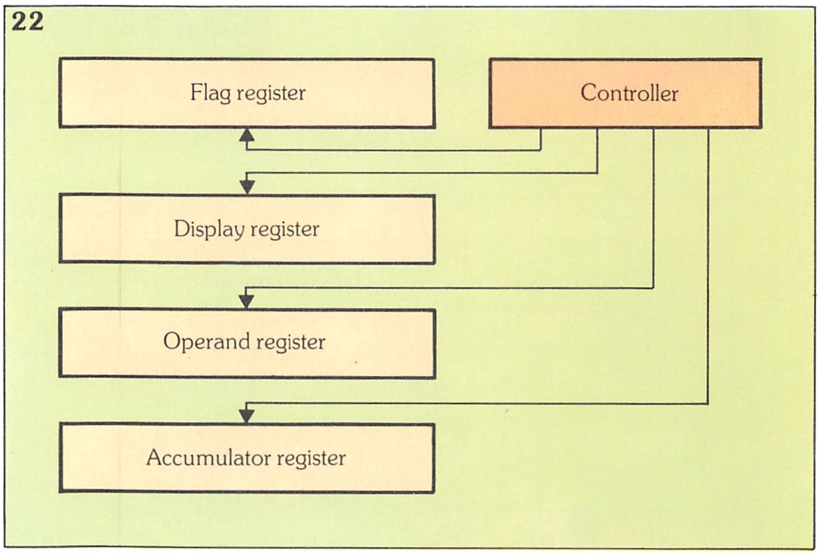
## What happens before beginning a calculation

Now we know the basic facts let's look again at the problem of adding 3 + 5. When the calculator is switched on it fetches the first instruction from the microprogram memory. This tells it to clear all information present in the 'register' subsystem, shown in *figure 22*. This clearing step eliminates any kind of useless random information which might be present when the system is first turned on. This takes place within the time of one instruction cycle, by means of a control signal sent to all the registers which are temporary memories for numbers and for other types of information.

In the next instruction cycle (instruction no. 1), the controller begins what is called the 'idle routine', carrying out an instruction which translated into ordinary language would say: 'Check if there are signals coming from the keyboard. If you don't see a signal, repeat the same instruction again. If you find a signal, go on to the next instruction'; i.e., the controller waits for any key from the keyboard to be pressed. Whenever the controller finishes a sequence of instructions it automatically goes back to the idle routine.

In the meantime, the **scan generator** subsystem (shown in *figures 18* and *19*), is working on its own without paying any attention to the controller; it is busy counting the clock signals and switching on and off one scan line after another (as we have already seen) at the beginning of each 100-microsecond instruction cycle.

The subsystem for **segment decoding** on the top right in *figure 18* is also working away on its own. Its job is to maintain the illuminated display, showing the numbers which are present at that moment in the display register subsystem,




**22. When the calculator is turned on** the controller sends a control signal to clear unwanted information from these storage registers.

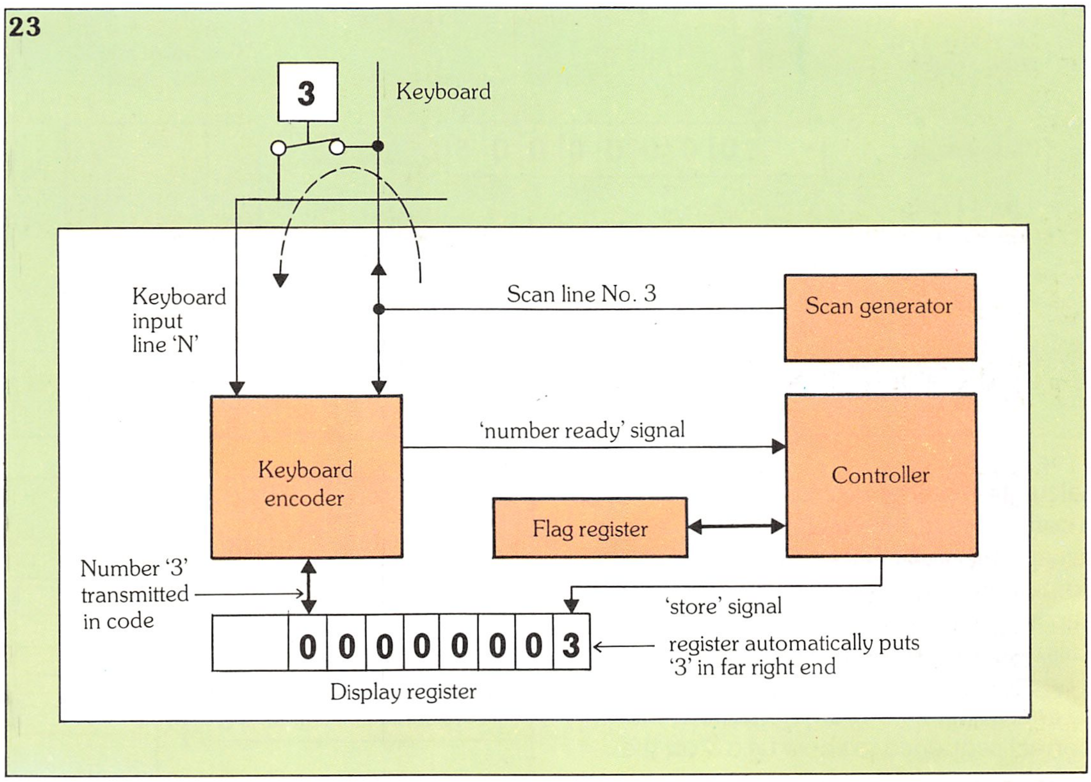
by activating the relevant segment lines so that they receive current at the proper times. The display register is a temporary memory circuit storing an eight digit number, complete with a decimal point and minus sign.

Every time that a new scan line comes into action, the decoder checks the relevant digit position in the display register, and deduces from that which segments are to be illuminated to show this digit in its position in the display. It also cancels every non-significant zero placed at the beginning of a number, except in the case where there are no numbers memorized in the register (blank register); in this case it shows one zero followed by a decimal point. This is, in practice, what you see before starting a calculation.

## What happens when you press key 3

Now let's look at precisely how the calculator handles a simple addition of 3 + 5. When you press key 3 (see *figure 23*), nothing happens until the scan generator activates scanner line no. 3. At this point a signal is sent from the keyboard access line N to the **keyboard encoder** subsystem. Knowing which is the active scanner line, this encoder generates the number 3 through a special code which can be transmitted electronically to the display register and memorized there.

The encoder also sends a signal to the controller, saying that a number key has been pressed. The encoder does not say *which* key has been pressed because there is no need for the controller to know this.



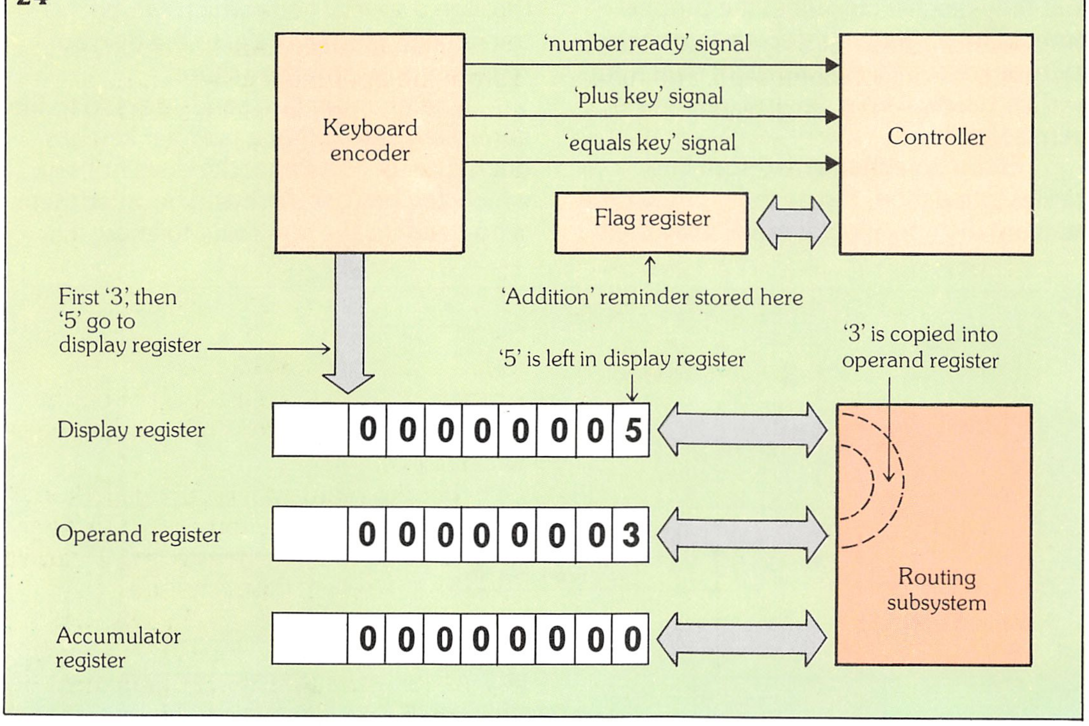

**23. An enlargement from Figure 19** showing the parts that are involved in entering number 3 from the keyboard.

To make sure that a key has actually been pressed and that the encoder has not reacted mistakenly to interference, the controller fetches and carries out a small series of test instructions.

Finding the 3 signal still present, (remember that the switch is much slower than the digital system), the controller has next to decide what to do with it. Still following the programmed steps, it checks whether there are any previously memorized indications in the **flag register** subsystem (see *figure 23*). These might indicate the type of problem to be solved and what steps in the problem have already been carried out.

At this moment, no indications of any kind are found in the flag register, so the controller leaves a sign there to remind itself that the first key to carry out a new

24



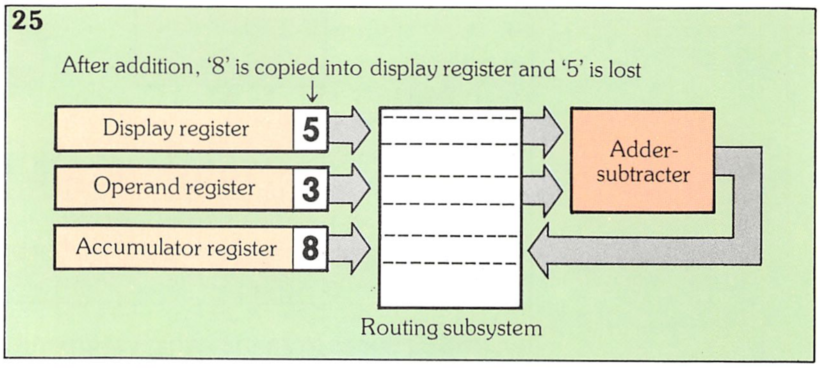

**24. Steps that occur in entering 3 plus 5** from keyboard.

calculation has been pressed. It then sends a command to the display register to accept the 3 which it has been seeing for some time. Since this register possesses separate places for eight digits, it automatically places the 3 in the far right position. The segment decoder immediately begins to energize the correct segments in the correct sequence to show us a 3 on the display.

Finally, after dozens of instruction cycles, as the controller methodically follows the instructions and decides which instruction to carry out next, our 3 appears and the controller can return to its idle routine. After making sure that all keys have been released, so that it won't enter another 3, it goes back to looking at the keyboard and waiting for another key to be pressed.

All this has taken only about one thousandth of a second. You are beginning to discover just how many different things must happen in a digital system, to execute even a very simple operation. Reading on you will see that speed is the true secret of success in digital electronics. Every job and every number is broken down into many tiny steps and elements (bits) which can be dealt with and processed by relatively simple electronic circuits. Thousands of these circuits assembled on a single integrated circuit chip can work together to process numbers and solve problems of the most complicated kind.

25




**25. Routing of numbers** during addition process.

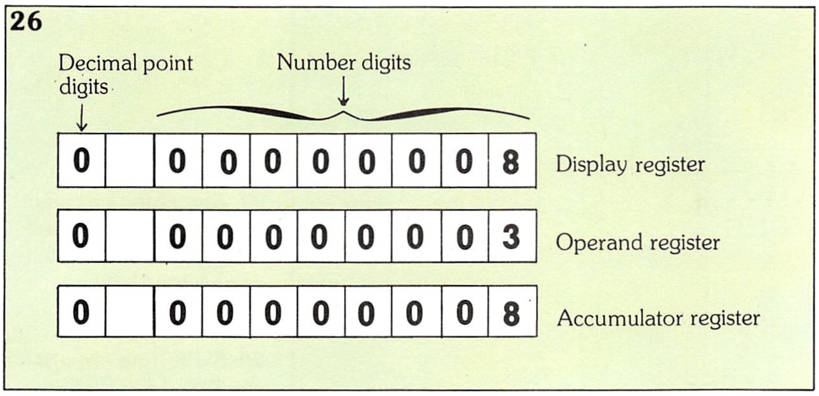
## What happens when you press keys + and 5

Moving more quickly through the problem of addition, look at *figure 24* which shows several more subsystems isolated from *figure 19*. When you press the '+' key, the encoder tells the controller that this has happened, and the controller in turn, as explained before, checks that a key has in fact been pressed.

Recognizing that this time it has received an addition command, rather

than a digit signal, the controller checks in the flag register to see if any previously entered functions have to be carried out before the addition. Not finding any, it follows the addition command, and makes the addressing subsystem place a copy of the 3 that is in the display register into the **operand register**. In other words, it transfers the 3 to the operand register where it will remain memorized.

The operand register is identical to the display register and the accumulator register. These will be further explained, but basically all three serve to memorize an eight digit number, with decimal point and minus sign if necessary. Pressing the 5 key triggers the same sequence of operations as previously when key 3 was pressed. The display register is cancelled and the 5 is deposited there, while the indication that a new number has been inserted is memorized in the flag register. So now the number 3 is memorized in the operand register, and the number 5 is in the display register (*figure 24*).




**26. Decimal points** in the number registers are handled by a separate digit.

### What happens when you press the = key

Finally, when the = key is pressed the encoder tells the controller which, in turn, checks and verifies the signal, acknowledging receipt of the end of calculation command (=). Recognising that an equals (or 'end of problem') signal has been received, the controller checks in the flag register to find out what operation it must carry out. It finds the indication that this is an addition. This leads the controller to a new instruction, the first of a sequence of program instructions called a **routine** – in this case an add routine (see *figure 25*).

Using several instruction cycles, the controller causes the addressing subsystem to transfer a copy of the 5 in electronic code from the display register and simultaneously transfers a copy of the 3 from the operand register. These two numbers go into the **adder-subtracter** subsystem where the addition then takes place.

The adder-subtracter subsystem is a unit which carries out all the arithmetical functions in the calculator. All that it can do (and as we shall see, all that it *needs* to do for multiplication and division as well) is exactly what its name suggests: add or subtract. The details of how numbers are added together electronically will be dealt with later on.

For the moment it is enough to know that, in a single instruction cycle, the 3 and the 5 are added together and the electronic code for the sum 8 is put into the **accumulator register**. Further instruction cycles work out the decimal point and the sign (+ or −) of the answer, then transfer them to the display register. Here, the usual display sequence makes the 8 appear on the display of the calculator. You aren't aware of it, but the 5 which was present in the display register has been cancelled and lost; however the 3 is still stored in the operand register, in case it should be required for other arithmetical operations. The controller now finds itself in the 'idle routine', waiting for other keys to be pressed, while the sum 8 remains on the display.

### How the decimal points are handled

There was another routine that the controller had to carry out to make sure that the numbers were added correctly. It had to check the position of the decimal points in the numbers that were added and make sure the adder-subtracter had them 'lined up' properly for addition.

In each register where a number is memorized, there is a digital code which indicates the position of the decimal point. The 0 in the decimal point box in *figure 26* is a code which indicates that the decimal point goes at the extreme right of the numbers memorized, i.e. the numbers dealt with are whole numbers. This coding system will be explained in more detail later.

# How electricity can transmit numbers

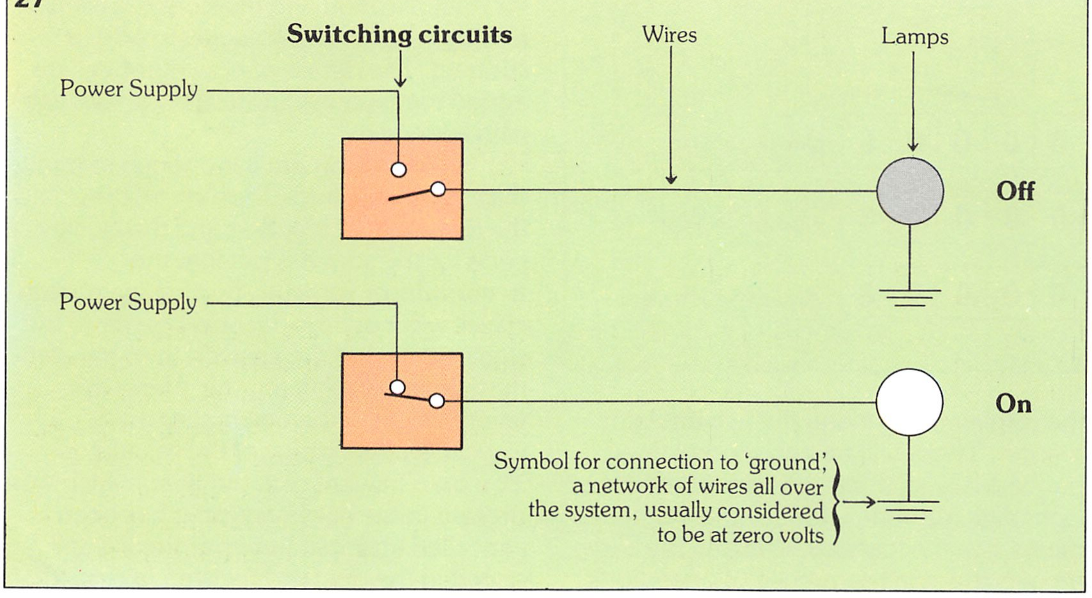
So far you have a general picture of how the calculator goes about its business. You've seen all the major parts of the system, and how they work together. All the possible complications – such as entering decimal points and minus signs – haven't been covered yet, nor have subtraction, multiplication, and division. But the fact is that all these matters are handled by the very same subsystems that have already been seen in operation. They are managed by appropriate steps, done one at a time, in accordance with programmed sequences of instructions.

For the moment let's move on to see more precisely how numbers are represented in a digital system such as this. We have seen where the 3 and the 5 ended up in the calculator IC, and we have stated that these numbers were encoded so they could be processed electronically. But what do they look like inside the IC?

Look at *figure 24* again, concentrating on the connection between the keyboard encoder and the display register. We know that the encoder generates an electronic code capable of representing the numbers from 0 to 9, corresponding to the ten number keys. On the diagram, a big arrow links the encoder to the display register and indicates a pathway for the numbers. What exactly is this path and how does it work?

The answer lies in the way a digital system breaks down all the jobs and all the numbers into many small steps and bits. This is so the tasks can be handled by relatively simple electronic circuits, which can easily be grouped together in large quantities on a single integrated circuit. In this way complex programs are managed.

27

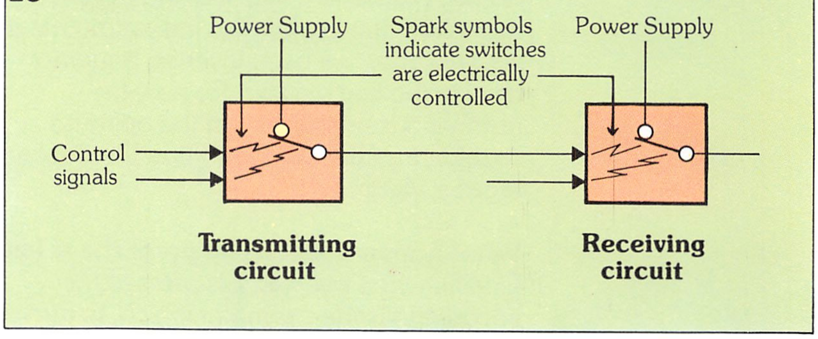



**27. Switching circuits** can be imagined as electrically controlled switches operating lamps.

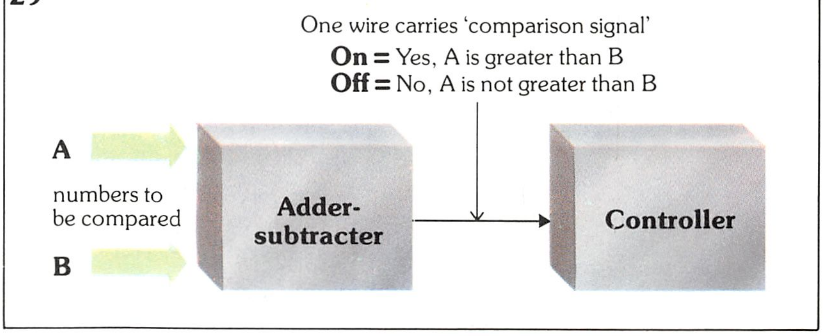
**28. Switching circuits can receive** as well as transmit switching signals.

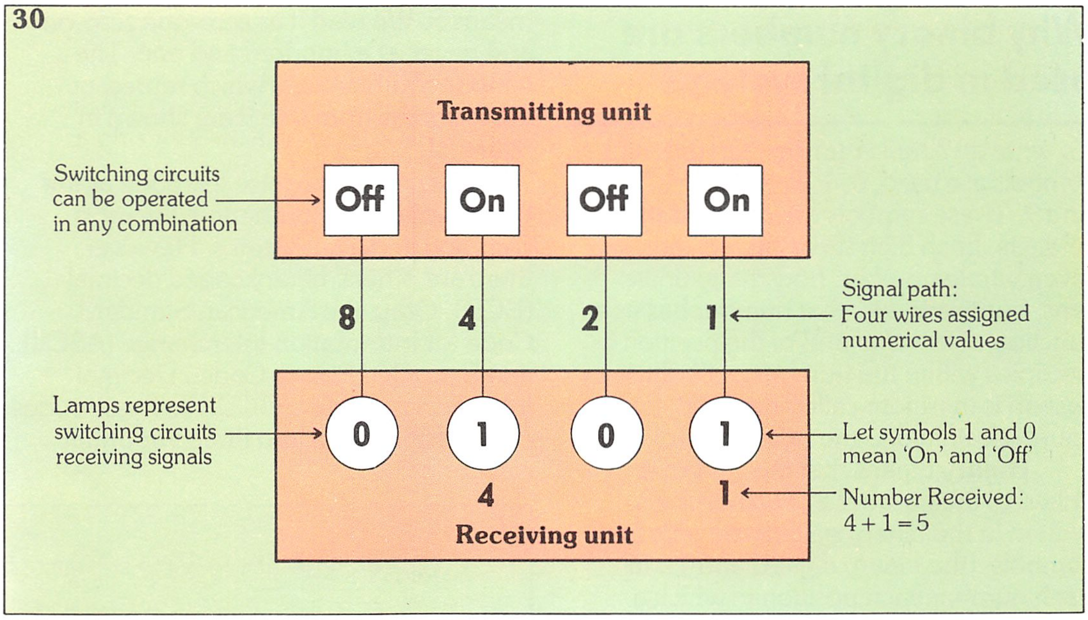
**29. Example of how much information one wire can carry** by being switched on or off.

28




29






**30. Transmitting numbers in a calculator** by switching several wires on and off. Each signal wire handles one bit of information. Each bit represents a decimal number which can be combined (added) by the receiving unit to make the number which was originally transmitted. In this case the '8' lamp is switched off, the '4' lamp is on, the '2' off and '1' on. This means that the number 5 was transmitted, represented as one '4' and one '1'.

## The simplest electronic circuit

The simplest sort of electronic circuit is one that just switches electricity on and off in a wire. Digital switching circuits are never partially connected. The lines controlled by them are always clearly in one of the two alternative states: connected or disconnected, open or closed, on or off, with a high or low voltage, a strong or a weak current, and so on. The switching circuits shown in *figure 28* are controlled by one or more input signals, which in turn are on or off. This means they can receive instructions from other switching circuits. This is how numbers and information are sent from place to place in every digital system – by switching circuits turning one another on and off.

## How information can be transmitted using a switch

How can such a simple device process the, at times, very complex information present in a digital system? Certainly a switch cannot say much, but it can say *something*. To give an example, consider *figure 29*. Two numbers A and B are 'compared' by an adder-subtracter unit to see if A is greater than B or vice versa (a problem which the adder-subtracter unit deals with by carrying out a subtraction).

By switching a single line on or off, the adder-subtracter unit can supply the answer to the controller. On means 'yes, A is greater than B', and off means 'no, A is less than B'.

This is an example of the fundamental unit of information in any digital system, the very simplest type of statement that can be made. It is merely a question of deciding between one of two alternatives, a simple matter of either yes or no. This amount of information is called a bit. It represents, as its name suggests, the smallest possible element of information. Therefore, at any one moment, a switch is able to transmit one bit of information.

But how can a switch capable of transmitting a bit of information solve the problem of transmitting whole numbers? What type of number can be transmitted with one single bit? 0 could be represented by the off state and perhaps 1 by the on state, but then what?

The answer is that to transmit numbers greater than 1, we simply use more than one wire instead of a single one. This gives a lot of combination possibilities and each combination can correspond to a different number according to a given code.

You can see in *figure 30* how numbers are transmitted in this 'example' calculator. The transmitting unit placed above, and the receiving unit below, represent the encoder and the display register of *figure 24*. But the same units can represent any two subsystems which transmit and receive numbers – taken at the very simplest level they all work in the same way.

## Why binary numbers are used in digital systems

To write an Arabic number, ten different symbols are used; 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. These symbols are called numerals or digits. Each digit serves to indicate, in a given whole number, how many units, tens, hundreds, etc. that number has as a function of its value and of the position it occupies within the number itself. This system is therefore called **decimal**, because it is linked to the number ten.

**Binary** means that the system is linked to the number 2. The numbers written in the binary system use only two symbols, (the binary digits 0 and 1). In each number each position in which a binary digit is placed expresses a value double that of the preceding position.

A digital system is one which uses digits to represent all types of information; even information that has nothing to do with numbers is reduced to a numerical form, using special codes which are also composed of digits.

But to represent numbers electrically in pure decimal form, you would need ten distinct voltage levels for each different digit, which would mean rather complex and costly switching circuits. Therefore it is preferable to use the binary digits (zero and one), which can be represented using very simple electronic circuits that can be switched on and off.

### An example of code used for numbers

As can be seen from *figure 30*, the code for the number taken as an example is found by letting each single line (wire) represent one of the following numbers: 8, 4, 2 and 1. The number transmitted corresponds to the sum of the numbers represented by the lines which are switched on. In the figure illustrated here, the wires indicated by 4 and by 1 are 'on', so that the number transmitted is 5.

Suppose that there is a way of indicating which line is on, for example the little lamps which light up, shown on the diagram. Under these lights you will see a row of 4 symbols: 0101. Assume that zero means off and one means on. Therefore, in this particular code scheme 0101 always means 5. We read it as zero-one zero-one and never as a hundred and one. The various combinations which represent the numbers from 0 to 9 are shown in *figure 31*.

You may recognise this code as the binary system of numbering, the most common in digital systems. However, there are others; binary coded decimal (BCD), Gray, the American Standard Code for Information Interchange (ASCII) and Extended Binary Coded Decimal Interchange Code (EBCDIC). These codes can be subdivided into three important categories:

**31**

| | **8** | **4** | **2** | **1** ←Value or 'weight' of each wire | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | = | 0 | + | 0 | + | 0 | + | 0 | = | **0** |
| | 0 | 0 | 0 | 1 | = | 0 | + | 0 | + | 0 | + | 1 | = | **1** |
| | 0 | 0 | 1 | 0 | = | 0 | + | 0 | + | 2 | + | 0 | = | **2** |
| | 0 | 0 | 1 | 1 | = | 0 | + | 0 | + | 2 | + | 1 | = | **3** |
| 0 = Off | 0 | 1 | 0 | 0 | = | 0 | + | 4 | + | 0 | + | 0 | = | **4** |
| 1 = On | 0 | 1 | 0 | 1 | = | 0 | + | 4 | + | 0 | + | 1 | = | **5** |
| | 0 | 1 | 1 | 0 | = | 0 | + | 4 | + | 2 | + | 0 | = | **6** |
| | 0 | 1 | 1 | 1 | = | 0 | + | 4 | + | 2 | + | 1 | = | **7** |
| | 1 | 0 | 0 | 0 | = | 8 | + | 0 | + | 0 | + | 0 | = | **8** |
| | 1 | 0 | 0 | 1 | = | 8 | + | 0 | + | 0 | + | 1 | = | **9** |

Binary numbers:
0 and 1 are binary digits or 'bits'

Decimal digits

**31. The binary number code** used to transmit numbers in our example calculator.

| Decimal number | Binary number |
|---|---|
| 1 | 1 |
| 3 | 11 |
| 7 | 111 |
| 15 | 1111 |
| 31 | 11111 |
| 63 | 111111 |
| 127 | 1111111 |
| 255 | 11111111 |
| 511 | 111111111 |
| 1023 | 1111111111 |
| 2047 | 11111111111 |
| 4095 | 111111111111 |

**Table 1**
Examples of decimal numbers translated into binary.

– codes used in electronic circuits to do digital jobs (e.g. binary)
– codes used to convert the whole decimals 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 into digital form (e.g. binary coded decimal BCD, Gray, etc)
– codes used to convert numbers, the 26

**Table 2**
Each octal number is represented by a pattern of three binary digits.

| Octal number | Binary number |
|---|---|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

**Table 3**
Four binary digits are required to represent a hexadecimal number.

| Hexadecimal number or letter | Binary number |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

**Table 4**
In BCD each set of four bits can represent one decimal digit.

| Decimal number | Binary coded decimal |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

letters of the English alphabet, keyboard symbols and operations into digital form (e.g. ASCII, EBCDIC and Baudot).

### Binary code

The simplest digital code is a two state or binary code composed of one logic state 0 and one logic state 1. In binary code, the decimal number 0 is represented by logical 0, and the decimal number 1 by a logical 1. How then can a binary code be used to represent decimal numbers higher than 1? The answer lies in generating numbers greater than 1 (to the base 10) through the use of a binary counting system, using different combinations of 0 and 1 on a base (or root) of 2.

For example, the binary number 11101 equals 29:

$$11101 = (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 29$$

You can see that according to its position every binary digit (0 or 1) has a different weight or power of 2:

$$2^4 = 16,\ 2^3 = 8,\ 2^2 = 4,\ 2^1 = 2,\ 2^0 = 1$$

*Figure 31* shows a table which enables us to convert a decimal number (from 0 to 9) to its corresponding binary number, and vice versa. Looking at *table 1* you can immediately see that a binary number of 12 bits can encode 4096 decimal numbers (from 0 to 4095) in binary code, while four bits can encode only 16 decimal numbers from 0 to 15.

The term bit comes from the first and last letters of the term binary digit and represents the fundamental information unit in all digital systems: the smallest possible information part. A bit can have one of two values: 0 or 1.

### Octal and hexadecimal codes

Two codes which are in very common use are octal and hexadecimal. The complete octal code is given in *table 2*.
The octal code is a binary code of three bits. The hexadecimal code, whose increasing use is linked to the spread of 16 bit microprocessors, is given in *table 3*. It is a binary code of four bits. The letters from A to F are used instead of the decimal numbers from 10 to 15 because these have another meaning in the hexadecimal system.

### Binary coded decimal

One of the most widespread digital codes for representing decimal numbers is the 8 4 2 1, otherwise known as **binary coded decimal** (BCD). Table 4 relates decimal

numbers to their equivalent binary codes. The BCD code is a simple binary code of four bits which starts at 0000 and finishes at 1001. Every decimal bit is represented by four binary bits as for the decimal number nine, shown below:

$$\text{decimal } 9 = \frac{2^3\ 2^2\ 2^1\ 2^0}{1\ \ 0\ \ 0\ \ 1}$$

Even if it is very handy to use, it is not very efficient; in fact to represent a decimal number such as 4095, 16 bits are required instead of the 12 of the straight binary system:

4095 = 111111111111 in binary
= 0100 0000 1001 0101 in BCD
4 0 9 5

### The ASCII code

It is not possible to limit the encoding and decoding of information to just numbers. It is equally important to be able to represent digitally the entire English alphabet, symbols such as !, $, %, *, ( ), +, =, and certain keyboard operations such as 'tab return', 'back space', 'shift', 'carriage return', 'space'.

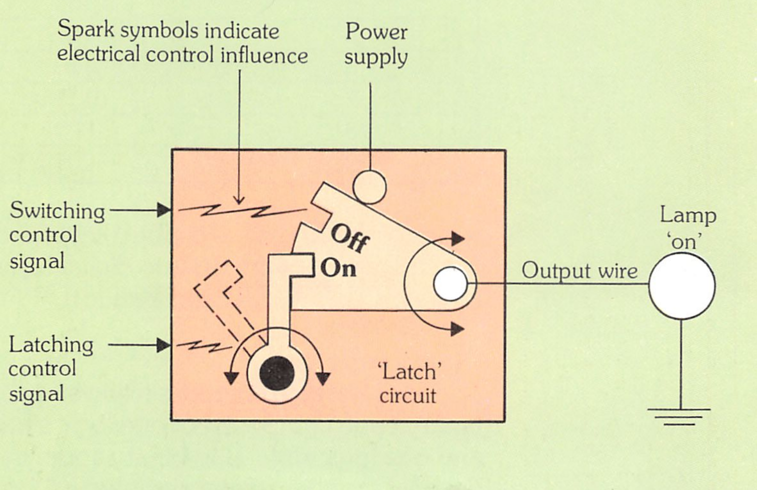
Alphanumerical codes are used to do this. The most widespread of these is the **ASCII** (American Standard Code for Information Interchange). The ASCII code is a seven bit binary code with a potential of representing $2^7$ (i.e. 128) different information elements. Half the ASCII table is shown here, that is, a list of 64 words in ASCII code with seven bits for the letters of the English alphabet, numerals and symbols. The remaining 64 words are keyboard operations.

As you can see, the ASCII code is even more inefficient that BCD in representing decimal numbers. For example, the decimal number 9 in ASCII code is 011 1001 and compares unfavourably with the 1001 of either the binary or BCD codes. However, maximum efficiency is not always the most important aspect of a digital code. Properties such as compatibility, utility and reliability can be more important in many situations.

Other widespread alphanumerical codes are the Selectric Code and the Extended Binary Coded Decimal Interchange Code (EBCDIC). Both are used mainly in IBM machines.

| Character | ASCII Code | Character | ASCII Code |
|---|---|---|---|
| @ | 100 0000 | space | 010 0000 |
| A | 100 0001 | ! | 010 0001 |
| B | 100 0010 | " | 010 0010 |
| C | 100 0011 | # | 010 0011 |
| D | 100 0100 | $ | 010 0100 |
| E | 100 0101 | % | 010 0101 |
| F | 100 0110 | & | 010 0110 |
| G | 100 0111 | ' | 010 0111 |
| H | 100 1000 | ( | 010 1000 |
| I | 100 1001 | ) | 010 1001 |
| J | 100 1010 | * | 010 1010 |
| K | 100 1011 | + | 010 1011 |
| L | 100 1100 | , | 010 1100 |
| M | 100 1101 | — | 010 1101 |
| N | 100 1110 | . | 010 1110 |
| O | 100 1111 | / | 010 1111 |
| P | 101 0000 | 0 | 011 0000 |
| Q | 101 0001 | 1 | 011 0001 |
| R | 101 0010 | 2 | 011 0010 |
| S | 101 0011 | 3 | 011 0011 |
| T | 101 0100 | 4 | 011 0100 |
| U | 101 0101 | 5 | 011 0101 |
| V | 101 0110 | 6 | 011 0110 |
| W | 101 0111 | 7 | 011 0111 |
| X | 101 1000 | 8 | 011 1000 |
| Y | 101 1001 | 9 | 011 1001 |
| Z | 101 1010 | : | 011 1010 |
| [ | 101 1011 | ; | 011 1011 |
| / | 101 1100 | < | 011 1100 |
| ] | 101 1101 | = | 011 1101 |
| ^ | 101 1110 | > | 011 1110 |
| ← | 101 1111 | ? | 011 1111 |

**32**



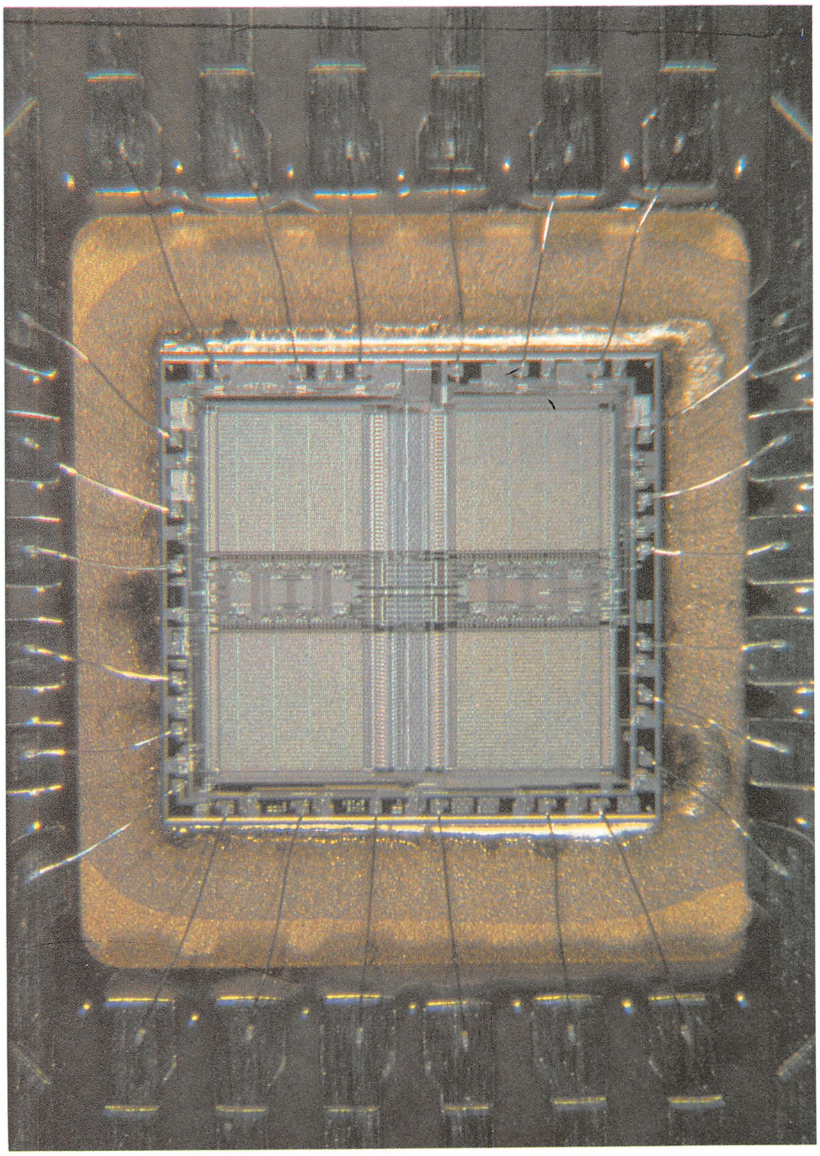
**Left: Table 5**

**32. Diagram of a simple switching circuit** that stores or remembers information.

**Typical memory chip.** The surrounding wires connect the chip to the pins of the package.

## How a switching circuit memorizes information

So far we have looked at the transmission, addition and display of information. The last function to be considered is information storage. If you have another look at the complete system illustrated in *figure 19,* you can recognize different subsystems which, as has been mentioned, serve to store numbers and information. The three numerical registers store numbers, the flag register stores different conditions, the address register stores the numerical addresses of instructions, and the instruction register stores the instructions themselves.



Science Photo Library

*Figure 32* represents a type of circuit called a latch which is a simple building block used in memory units. The name refers to the fact that the output can literally be latched in one state or the other – on or off. This has been indicated in the diagram by showing a mechanical latch engaging either of two notches (labelled on and off) in a pivoting mechanical switch element. Thus, the switch can be latched either in the on position – so that electricity flows from the power supply to the lamp – or in the off position. In reality, electronic latches use transistors rather than mechanical parts but the effect is essentially the same.

Let us see what happens when the latch lever is in the withdrawn position shown by the broken line. (Spark symbols are used to show the control function). The switch element is turned on or off by the switching control signal which comes from outside the system. This comes from an external switching unit which is attached to the connecting wire marked 'latching control signal'. With the latch withdrawn, this element switches each time the control signal changes. When, however, the latch is engaged, the switch can no longer change state and remains in this position until the latch is withdrawn once again.

So, to remember whether at a certain moment the switching control signal was on or off, in other words to store this bit of information, we simply latch the circuit into that state at that moment. The switch remains fixed in that state, regardless of how the switching control signal varies afterwards, until the latch is withdrawn again. As soon as the switch control signal modifies the state of the switch, the information which had been stored is lost or forgotten, as the output now represents new information to be labelled.

If this system is represented as a digital electronic circuit that is receiving one of the wires from the transmitting unit shown in *figure 30*, then putting four similar circuits together gives us a memory unit with a four bit capacity, which can be used to represent the numbers from 0 to 9, as shown in *figure 31*. All that's needed is a latch circuit with the relative control line in the four state receiving unit which constitutes a register. It can memorize four bits until they need to be modified.

## General principles that apply to all digital systems

From studying this 'example' calculator we have learned a number of important general principles which apply to digital systems in general. Firstly, all modern electronic digital systems work in the same way as the calculator, by reducing information and problems to very simple terms: on or off, yes or no, 0 or1.

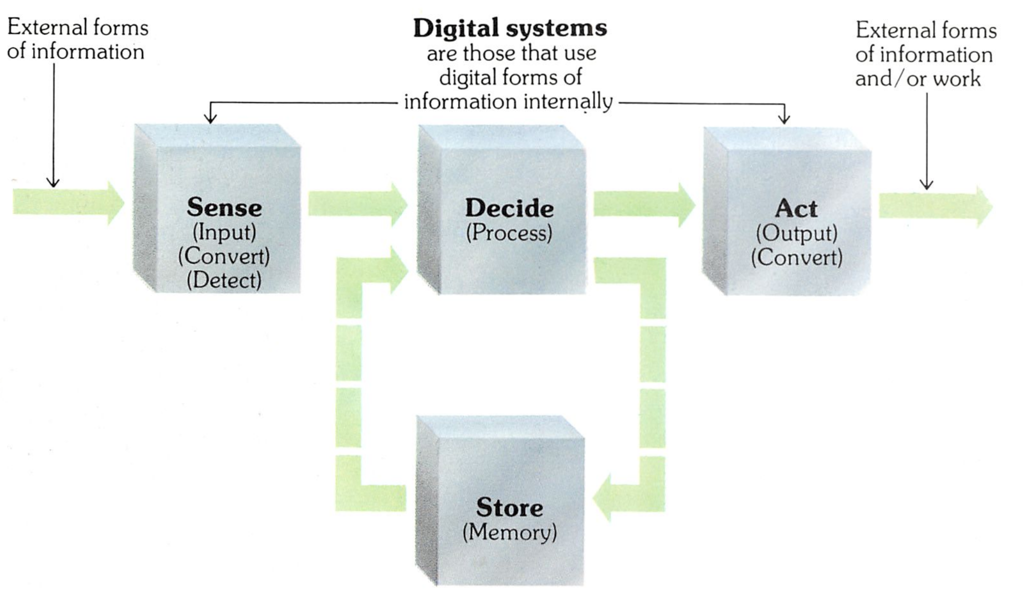
Large quantities of these simple units are required to handle information and tasks of any complexity. The work is accomplished rapidly, using code schemes by which many simple pieces of information can represent a more complex bundle of information. This is the pattern you will see operating in every digital electronic system.

### Why electricity is particularly suited to digital systems

It is possible to build a perfectly functioning digital system without using electricity in any way. Nothing in our definition of digital systems says anything about electricity, just the division of information into elementary units, the use of numerical digits and so on. We have already mentioned an example of a non-electrical system, namely the mechanical calculator.

The reason that electricity has been employed for digital systems so successfully is that electrical switching circuits – which are relatively simple and inexpensive compared to some other electrical circuits – can be used effectively to handle the very simple units of information involved in binary digital systems. These circuits offer the fastest, most convenient method so far developed for this task.

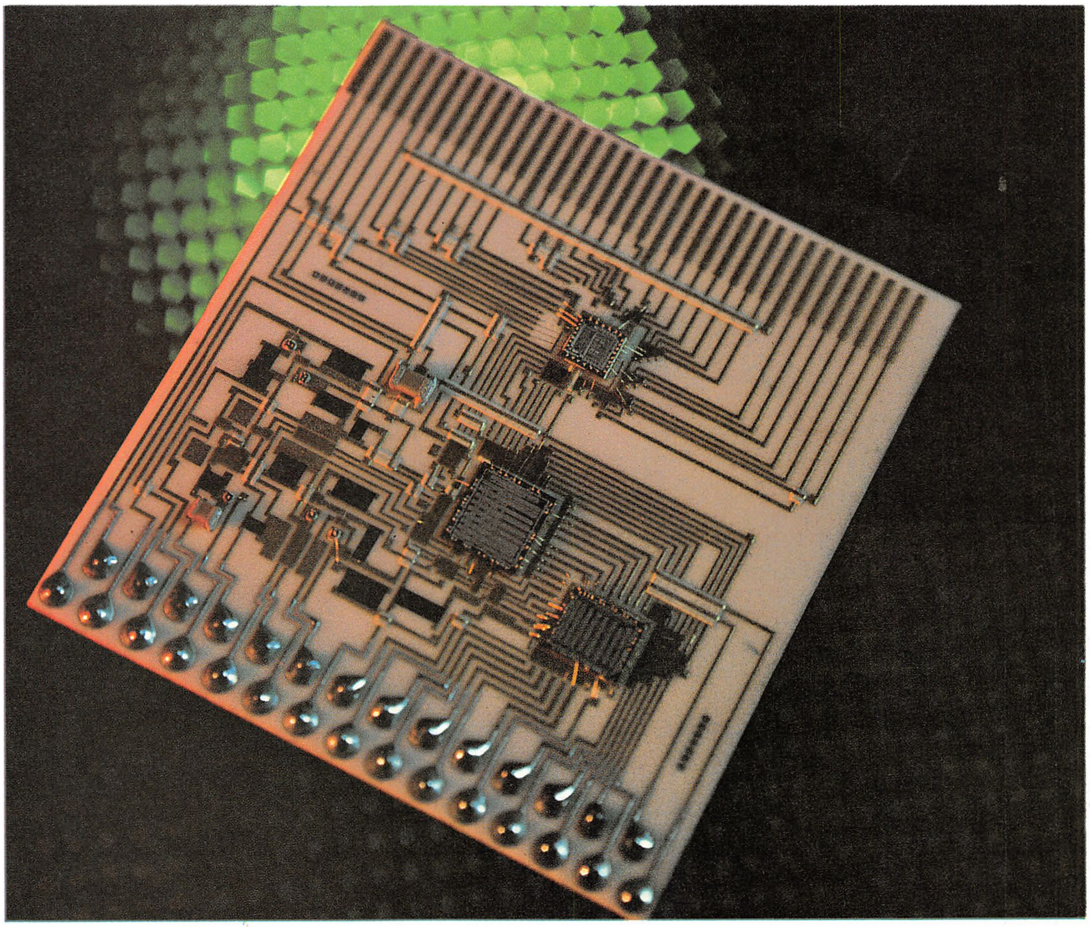
33




**33. How universal system organization** applies to digital systems such as the calculator.

### Why integrated circuits fit the purpose so well

The first digital electric systems used electromechanical relays that actually contained little mechanical switches similar to the imaginary sort used in our diagrams of switching circuits. Later digital electric systems used thermionic valves instead. Soon the transistor came along as a replacement and then semiconductor integrated circuits. These, by their very nature, are capable of reducing simple switching circuits to microscopic dimensions and can pack countless thousands into an incredibly small space, thus reducing considerably the cost per circuit. These are all characteristics which make integrated circuits perfect for digital systems, which involve many, many simple



Paul Brierley

**A network of integrated circuits** mounted on a thick film substrate.

tasks and small pieces of information.

### What all digital systems do, and how they are organized

Another, even greater, generalization can be drawn from the calculator example illustrated in *figure 33*. This generalization is made up of two ideas.

Firstly, the only tasks that any electrical system can do are either to manipulate information or do work (or both). Second, all systems are organized in the same way. They carry out their jobs of handling information and/or doing work according to the same basic pattern of steps. This is known as the Universal System organization.

Initially, they **sense** (detect or accept) information in varying forms coming from the external world, and convert it into a form which can be handled within the system. Then they take **decisions** based on this input information – in other words, they process or manipulate the information in some way.

In doing this they may **store** or 'remember' some parts of the information for a certain length of time, or process it as a result of other information which is permanently stored in the system's memory.

Finally they take the information which results from this processing and reconvert it to a form which is comprehensible to the external world, perhaps in the form of output which exercises control over operations or machines – in any case, with some kind of **action.**

Think of any digital system and you will see how this universal organizational scheme can be applied. For example, the keyboard and the coder of our pocket calculator 'sense' the information and convert it into a suitable internal form. Various subsystems then 'decide' and 'store'. Finally, the segment decoder and the display system convert the information that results from this internal processing into the required 'action' to show the numbers of the answer on the display.

# Glossary

| | |
|---|---|
| **analogue** | a method of transmitting information in an electrical circuit, by means of varying the current or voltage |
| **ASCII** | American Standard Code for Information Interchange. Seven bit binary code used for the inter-machine exchange of alphanumerical data |
| **binary** | number system linked to the base number 2. The numbers written in the binary system use only two symbols: the binary digits 0 and 1 |
| **binary code** | system of representing numbers using binary digits |
| **chip** | a tiny piece of semiconductor material on which various components such as transistors, diodes, etc., are miniaturized by means of varied technological procedures |
| **clock generator** | the timing subsystem |
| **clock signals** | timing impulses used for synchronizing all the operations of the system |
| **controller** | the part of a programmed system which has the job of telling each subsystem when it must act and what it must do |
| **digital** | a method of transmitting information using only two values, on or off (0 or 1, true or false) |
| **hexadecimal code** | a 4-bit binary code linked to the base number 16, which can be used in the control of 16-bit microprocessors |
| **idle routine** | a continuous checking routine, watching for incoming signals from the keyboard, carried out by a calculator controller while it is not doing any work |
| **integrated circuit** | complete miniature electronic circuit, whose components are fixed on a semiconductor chip |
| **keyboard encoder** | subsystem which informs the controller that a key has been pressed; the controller in turn will check which key this is and decide what action to take |
| **LED** | a light emitting diode |
| **octal code** | a three bit binary code linked to the base number 8 |
| **register** | a type of temporary storage unit used to hold digital information until it is needed |
| **scan generator** | subsystem which controls the switching, at defined intervals, of the scan lines in a digital circuit |
| **segment** | in digital electronics, 'segment' refers to one of the seven bars of a rectangular figure-8 pattern in a character display |
| **subsystem** | a smaller system inside a larger system. Each subsystem can be thought of as a separate system with its own job to do |

2

SOLID STATE ELECTRONICS

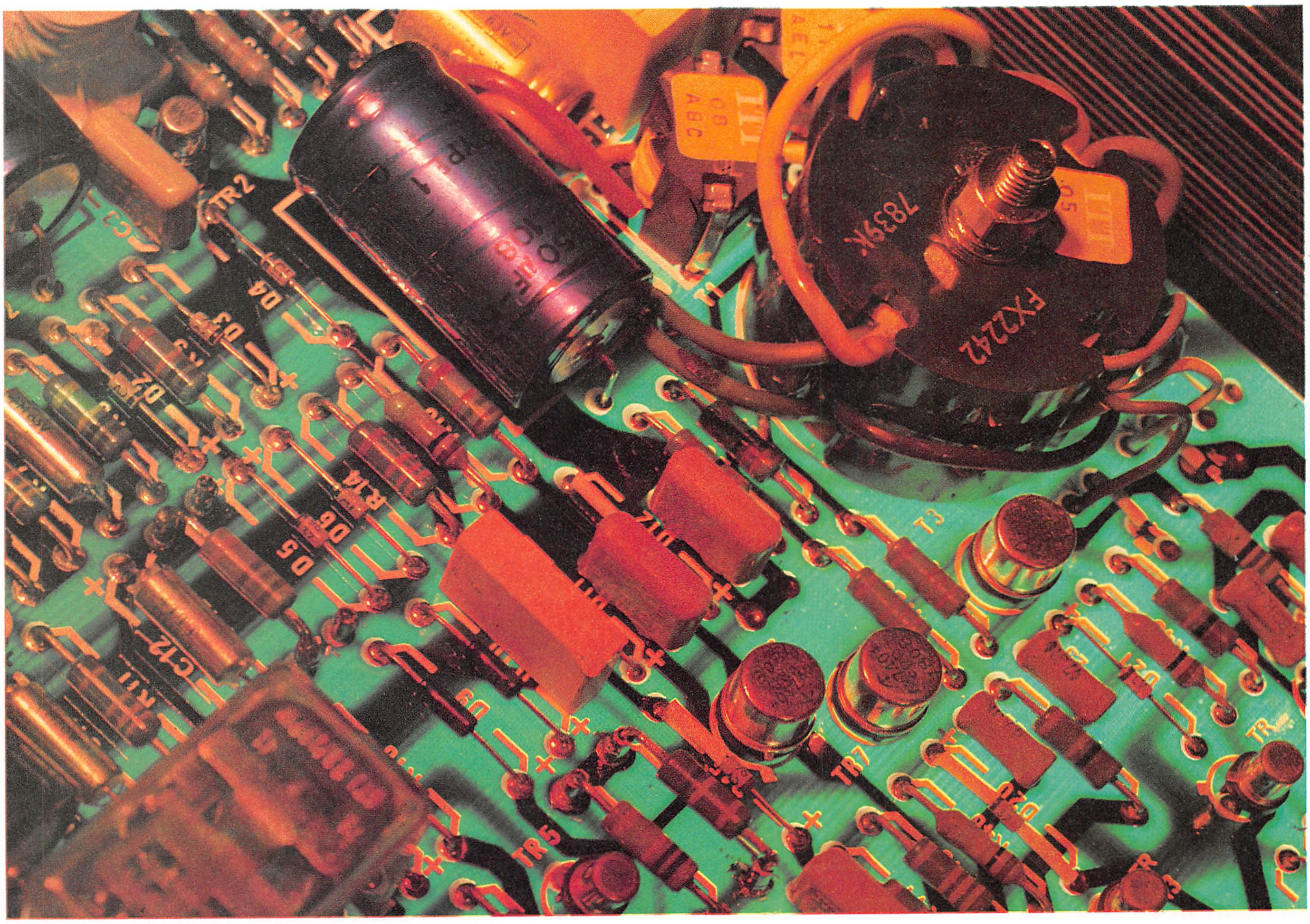
# Basic circuits and their functions

## Basic switching and amplifying circuits

In chapter 1 we analysed the characteristics which are common to all systems: the next step is to look at circuits. As this will involve jumping to and fro between the various levels on which a system is organized, it's important to keep these levels clearly in mind. The highest organization level is the **system** (a radar system, a television, a watch, a radio); within each system there are three **stages** (sense, decision and action); within each stage are one or more **circuits** (tuning circuit, meter circuit, light sensing circuit); inside each circuit there are one or more **components** (transistors, diodes, rectifiers, integrated circuits, resistors, capacitors). So although a particular system has only three stages, it can have thousands of circuits and millions of components.

Much of what you'll be dealing with is based on the single fact that there are only two things which can be done to electricity between a power source and a usage point: it can be **switched** or **varied**. Some elementary methods of switching and varying have already been covered in the

**Below: A pre-integration circuit board**; note the use of transistors.




Paul Brierley

previous chapter. Electrical power, for example, can be regulated by a variable resistor. A common type of variable resistor is the **potentiometer**. By turning the knob on the potentiometer, the amount of resistance can be varied. This has the effect of reducing or increasing the brightness of a light or controlling the volume of a radio.

It's been explained, too, how a hand operated switch can be used to transmit telegraph messages. However, it's evident that manual switching and regulation are totally unsuitable for modern electronics. How can a system be practical if you need to manually switch and vary thousands of different circuits? Of course, the answer is that you can't build any workable system which is even slightly sophisticated using manual controls.

The invention of the vacuum tube or **valve** was a great step forward; it meant electrical power could be controlled by electrical means instead of using mechanical or manual methods. This meant that switching or varying could be carried out at high speed – millions of times a second.

### The importance of transistors

The invention of the **transistor** has, in turn, opened up far greater possibilities. Today it is at the heart of all modern electronics. The transistor carries out the same functions as a valve: it switches and varies using electrical means. In comparison with the valve, however, it has many advantages: it doesn't need a large standby operating current, it is very small and light, and it is tough and long lasting. Although able to conduct relatively high currents, it operates at conveniently low voltages, and its dependability is thousands of times greater than a valve.

To understand how circuits function, you need to see first of all how their basic component, the transistor, behaves and what it does in the circuit. *Figure 1* shows the most general and basic circuit arrangement. It has a power source, a work device and a control (which switches or varies current or voltage).

The heart of every transistor is a small piece of semiconductor material, usually silicon. As can be seen from the cross-section of a transistor shown in *figure 2*, there are three distinct sections or regions, either of 'P' or 'N' type. (The terms P and N will be explained later.) This transistor is of the NPN type. This semiconductor device can be made to act as a variable resistance or as a switch, ie. it conducts current, partially reduces its flow or blocks it completely. Look first at how a transistor can be used to *vary* the power in a circuit.

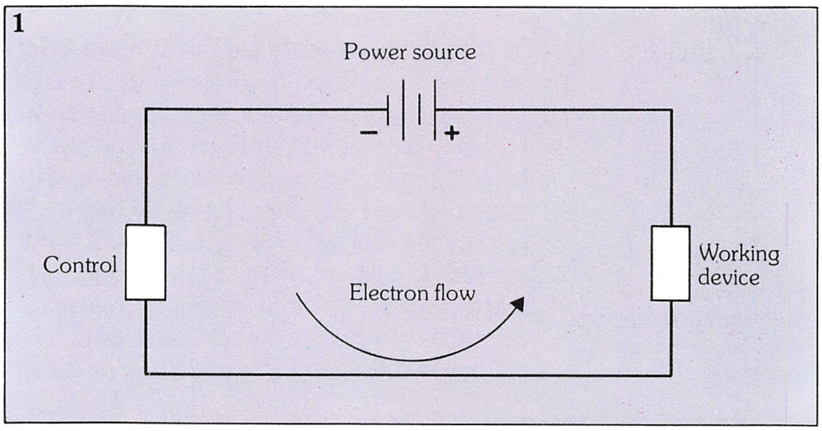
You can see how this happens by putting the NPN transistor into the basic diagram of the circuit, as shown in *figure 3*. The source of power is still a battery and supposing that the circuit will be used to power a loudspeaker, a microphone has been connected to the P region of the transistor.

**NB** Don't build this circuit, or any others in this section, for practical purposes. They have been simplified to make them easier to explain and certain components are missing.

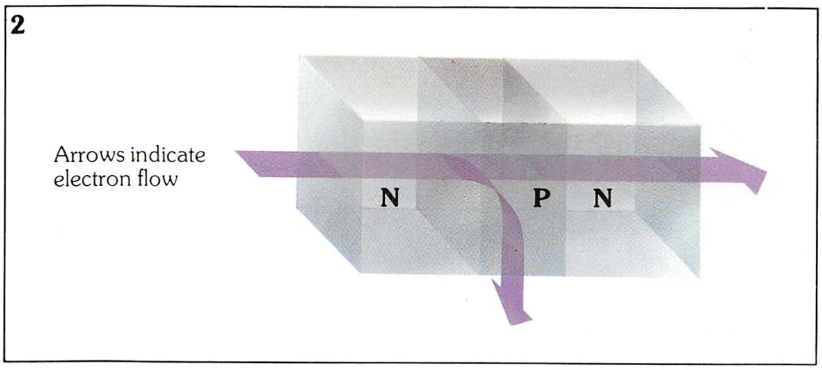
As long as the microphone is disconnected from the circuit, nothing happens. The transistor simply blocks the flow of electrons coming from the battery to the loudspeaker. To make current flow, electrons must be removed from the central region of the transistor, known as the **base**,

**Right: Part of a typical transistor circuit.** This one is actually from a smoke-detecting device.

1

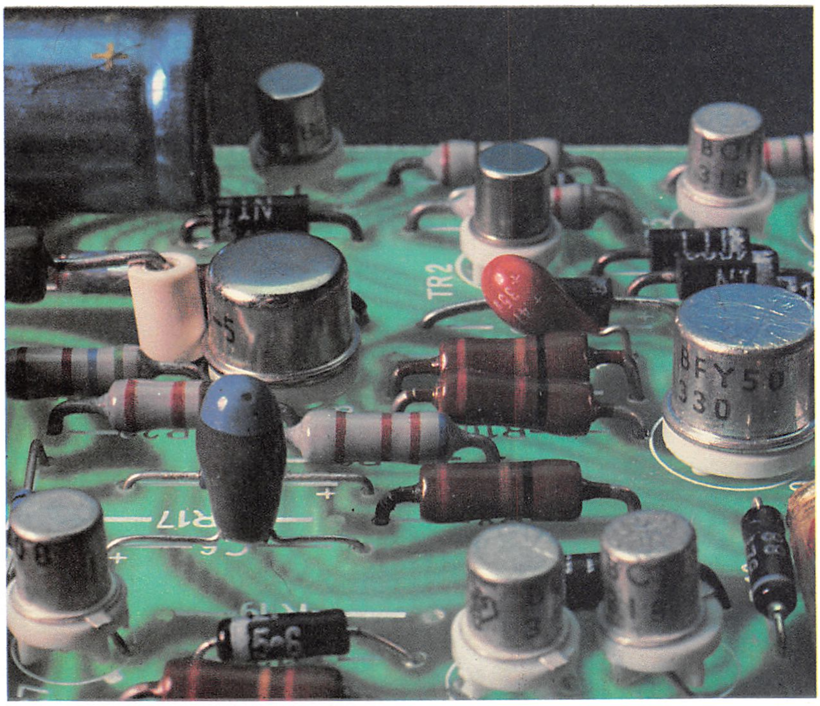



2

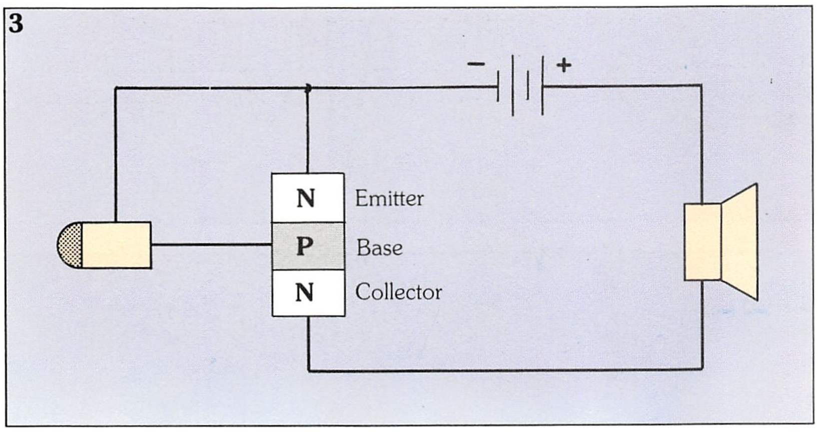



**1. General configuration** of an electrical circuit.

**2. Cross-section of a bipolar transistor.**



Science Photo Library




**3. Current control** using an NPN transistor.

to cause the current itself to move from one N region to another. The more electrons removed, the more current will flow.

One N region is called the **emitter** because, when electrons are removed from the base, this region will *emit* electrons across the base region. The other N region is called the **collector** because that is the region where the flowing electrons are *collected* before passing along the wire to the loudspeaker.

You may notice that in *figure 3* a wire has been connected from the microphone to the transistor's emitter. This is necessary to give the electrons removed from the base a place to go, back to the emitter. You'll remember from chapter 1 that electricity flows in a circuit *only* if it has a place to come from and somewhere to go to. This added wire has completed what is called the **control circuit**.

The proper symbol for a transistor can now replace its section diagram, as shown in *figure 4*. In this symbol the vertical line represents the base, the plain diagonal represents the collector and the diagonal with the arrow represents the emitter. The direction of the emitter arrow always points in the opposite direction to the electron flow. The symbol can be completed by a circle, but has the same meaning without it.

## How a transistor functions as an amplifier

Transistors which act as variable resistors are called *amplifiers*. In *figure 4* you can see how a transistor amplifier works. In this example the electrical control will be obtained by means of a microphone. When a sound wave hits a microphone a tiny part of the energy in the soundwave is absorbed, creating a potential difference at the microphone terminals. If the microphone is plugged into a circuit a small current will then flow which fluctuates as the sound waves vary. But the microphone can only produce a tiny source of electrical power.

If the microphone leads are connected directly to a loudspeaker, the current generated even by loud noise would be too weak for you to hear anything. Using the simple circuit shown in *figure 4*, you could, on the other hand, produce a sound loud enough to wake the neighbours.

To give a better idea, assume that the microphone can supply a power which varies from 0 to 5 mW (mW means milliwatts, i.e. one thousandth of a Watt). But the power produced by the battery in the main circuit can go from 0 to 500 mW. Now suppose that a single sound wave strikes the microphone and creates a power output of 3 mW. The microphone causes a rush of electrons to flow from the emitter and through the base. Now, as a result of the base current, a relatively powerful current crosses the base region from the emitter to the collector and continues along the line through the coil of the loudspeaker. In this way, the flow of current in the amplifier becomes controlled

or amplified in exact proportion to the much weaker signal of the microphone. The signal which passes through the loudspeaker could have a typical value of 300 milliwatts; which means that the 3 mW of power produced by the microphone has been amplified one hundred times.

Now suppose a second sound wave hits the microphone. This is a softer sound, and produces a power output of only 2 mW. Fewer electrons flow in the control circuit, so fewer are drawn from the base region, and this time the amount of power flowing through the transistor and the loudspeaker is only 200 milliwatts. Nevertheless, it has also been amplified one hundred times.

In all cases the power in the work circuit will ideally be a replica of the power in the control circuit, but very much amplified. You can visualize the process with the voltage trace illustrated in *figure 5*. If the sound produced by the microphone is the small squiggle on the left, the sound of the loudspeaker signal will be an exact copy of the small one but greatly magnified.

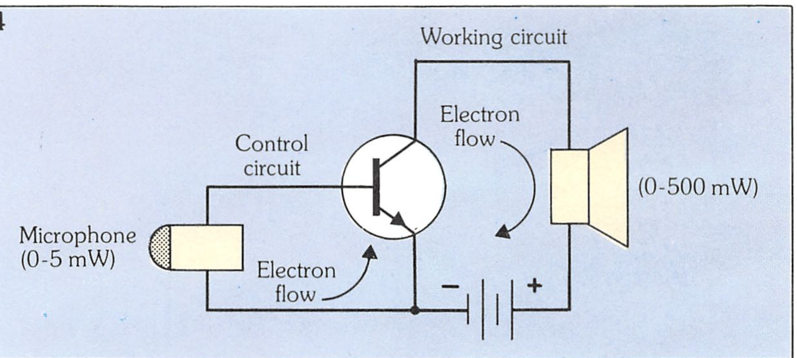
Speed of reaction is one of the transistor's most important characteristics, one that makes it very useful in modern electronics. Sound waves fluctuate very rapidly, at frequencies up to 30,000 cycles per second (30 kHz). The transistor is able to react to each one of these rapid fluctuations. In fact high frequency transistors have the capacity to react millions of times every second.

**How a transistor operates as a switch**

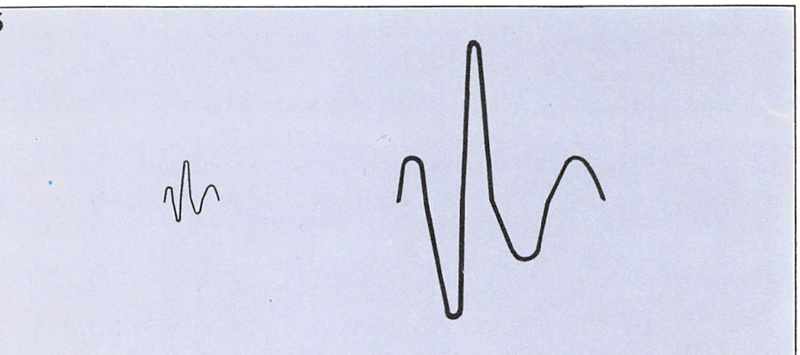
To show how a transistor can act as a switch, we'll make use of the simple telegraphic circuit shown in *figure 6*. This has a battery as its power source, a buzzer as the output, a transistor is the work circuit and in the control circuit a switch takes the place of the microphone. Since the switch can't generate electricity a battery is included in the control circuit. The zig-zag line represents the resistance given by sixty miles of wire. This resistance reduces the electron flow at the end of the wire, so that at a distance of sixty miles it is not enough to operate the buzzer, but it is sufficient to control the transistor. When the switch is pressed there is a relatively small flow of electrons from the base, causing a much higher current to flow in the output of the transistor so that the buzzer works. This transistor is operating as a switch in the work circuit.

Any transistor can either switch the

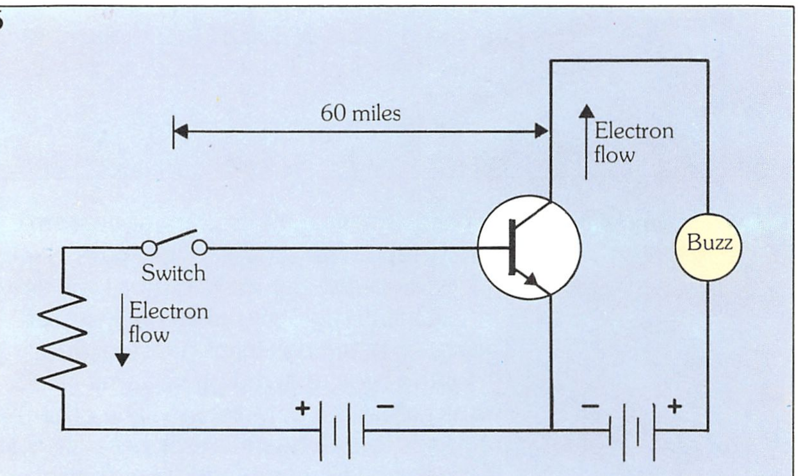
4




5



6




current in the work circuit on and off or, alternatively, vary the current from zero up to the maximum possible value. In other words a transistor works rather like a tap in an electronic circuit. It can be completely open, completely closed or set at any point in between. The tap is operated by varying the control signal (known also as the base

**4. The circuit used in figure 3** with the standard symbol for a transistor inserted.

**5. The voltage signal** coming from a microphone before and after amplification.

**6. Using a transistor as a switch** in a simple telegraph circuit

current). This has to exceed a certain minimum threshold level before the tap 'opens' and current flows in the work (or collector) circuit.

### The control circuit

What determines whether a transistor acts as a switch or an amplifier? Basically this depends on the control circuit. For example, compare the control stages of the circuits shown in *figures 4* and *6*. In *figure 4* the microphone produces a varying voltage on the base which causes a small current to flow from it, which in turn causes the current in the output to vary between zero and maximum.

In *figure 6* on the other hand, the control circuit is made up of the battery and switch. When the switch is open the base is below the threshold level and no current flows – the transistor is off. When the switch is closed current flows from the base because the battery is above the threshold level, and as a result current flows in the output circuit. As the battery voltage is constant the base current, and thus the current in the work circuit, will be constant. So the transistor operates just like a switch in the work circuit, switching the current on and off. (Normally the transistor chosen as a switch is one that allows its maximum current to flow when the control signal is applied. The technical term is that it is **bottomed.**)

In *figure 4*, then, you can see that the microphone in the control circuit causes the transistor to vary the current in the work circuit, while in *figure 6* the switch and battery cause the transistor to switch the current on and off.

Although each transistor can operate as both switch and amplifier, normally they are constructed to do one thing better than the other. Later on you will see mention of some transistors as amplifiers and others as switches. Most good transistor amplifiers have a moderate and stable current gain (the amount they amplify by), while those to be used as switches must be able to switch on and off very quickly and have very low losses. Since most digital computers are predominantly based on switching transistors you can well imagine the importtant role played by transistors in today's most advanced electronics.

## More complex circuits

The description of the switching and amplifying circuits was based on two simple and very basic circuits. If you actually built them you would find their performance very disappointing. A single transistor is the only semiconductor component in either of them and even the simplest of portable radios has at least six or seven transistors in it. But in addition to this, a typical system consists of other components besides transistors; diodes, resistors, capacitors and inductors are all common circuit components. What we need to do now is to go back to the basic circuits and add things to them to make them work better. In essence this means that they'll be more like typical electronic circuits. Not all the new components will be explained fully as they come along. But don't worry if this means you don't understand every circuit that is introduced. The idea at this point is to give some appreciation of the ways semiconductors are used in various circuits to achieve different results.

*Figure 7* shows a circuit which is more complex than any of the others seen so far. However, it is still easy to recognise the part of the circuit that does the work (on the right) and the control part on the left. The power source has been marked with a 'G' (standing for generator). This could be the mains socket in your home, a solar cell in a satellite, a battery or anything else that will supply power. There is a second power source on the left, also marked with a 'G'. This could easily be the microphone or battery and switch used in the previous examples. The device which does the work has been marked with an 'M', standing for motor.

Suppose that the motor has to be kept running at half speed even when there is no control signal from the generator; that's to say the motor has to be controlled from half speed up to its maximum speed. This can be achieved by adding a resistor to the control circuit to ensure that enough base current will always be flowing to keep the current in the work circuit high enough to run the motor at half speed.

When the control generator causes the base current to increase, the current in

the work circuit will also increase and the motor will be driven even faster. Another resistance has also been added to the work circuit. This is to stop the motor from overheating if the generator supplies more current than the motor can stand. It works by limiting the amount of current that can flow from the generator and through the motor. Supposing that the generator is an AC source rather than DC, it is also necessary to add a **diode rectifier** to the circuit. This device allows electrons to flow in one direction only, converting an AC current into DC. The arrow of the diode points in the opposite direction to the electron flow.

Finally suppose that the motor is very big and needs a lot of current from the control transistor. If the control generator is a very weak source it is likely that the low current in the base would not cause enough current to flow in the work circuit. The answer to this is to use a low power transistor in the control circuit and have it control a second high power transistor. Now, the current that is always flowing in the first transistor (because of the bias resistor) causes a much bigger current to flow in the second transistor. When the control generator causes the current in the control transistor to increase, an even bigger increase is felt in the power transistor and the motor turns faster. This explanation should make it clear how transistors can be used in different ways to do different jobs. The basic circuit function hasn't changed from the first amplifier described; the various parts just make up an amplifier that, in this case, is used to vary the speed of motor between half and full speed.

There are many other components that have not been mentioned yet which can increase circuit complexity even further. In later chapters these will be introduced and the way they work in both switching and amplifying circuits will be explained.

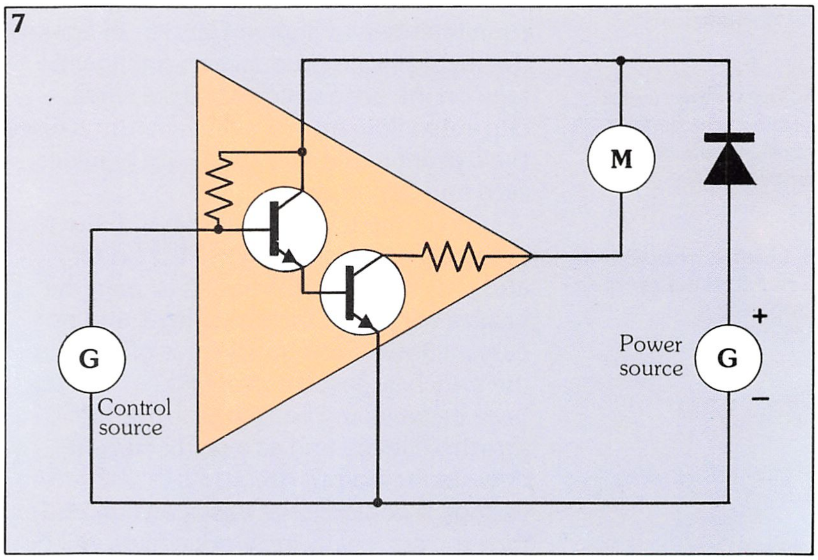
## Simplifying the circuit diagram

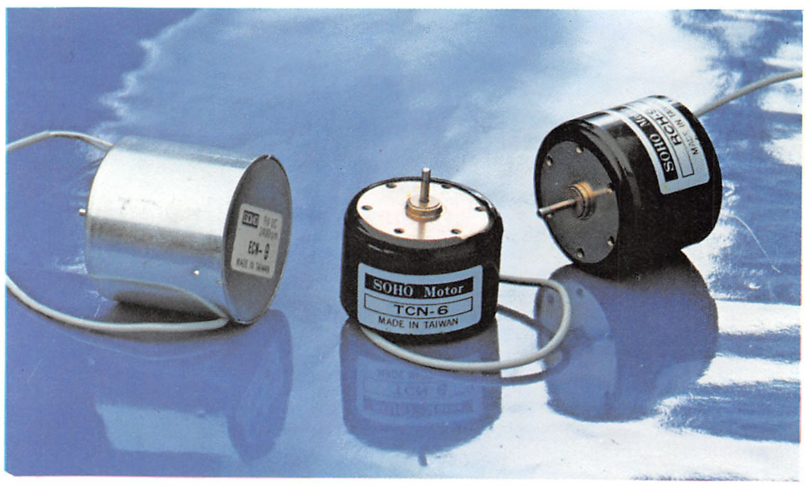
There is a simple way of analysing a system that can be called the block schematic method. Often each stage of a system consists of so many sense, decision and action circuits that at times it is difficult to understand how each part of the circuit is supposed to act within the system. In practice, however, it is normally sufficient to consider the system as a set of elementary boxes or blocks.

Returning to the motor circuit, for example, all the circuitry in the big triangle can be considered as a single amplifier, without worrying about all the components that are used. The symbol for an amplifier is a triangle, as is shown in *figure 8*. This circuit is functionally identical to the one shown in *figure 7*.

Control signals are shown entering the amplifier on the left of the triangle, the output to the motor is on the right and the power supply lines enter at the top and the bottom. These connections are common to all amplifiers and, knowing that it is an amplifier, you don't have to worry about

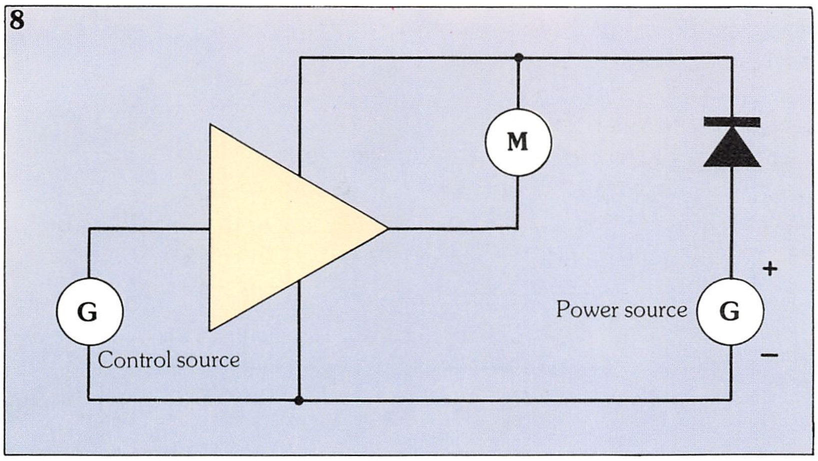
**7. A more complex circuit** to control the speed of a motor.



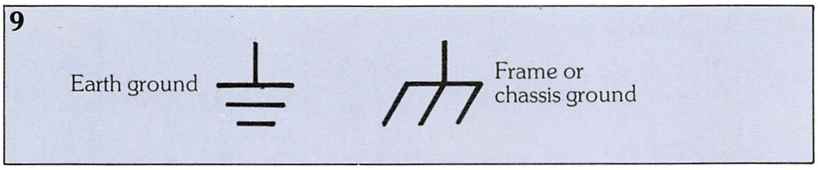




**Three small DC (direct current) motors.**

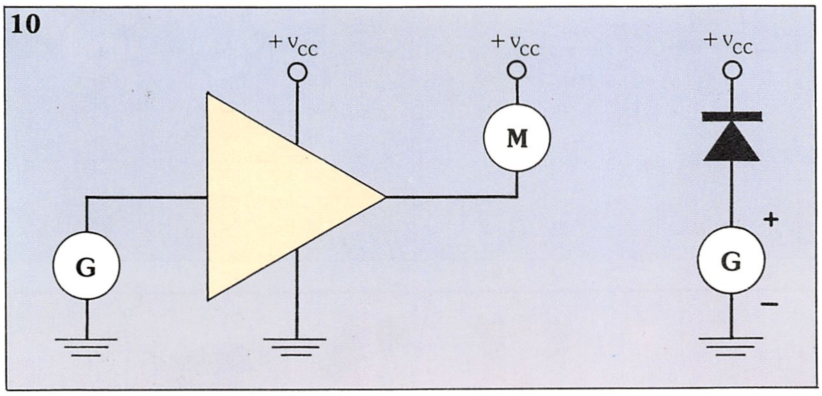
8




9




10




**8. Figure 7 circuit with the building block symbol** for an amplifier inserted.

**9. Conventional symbols** for representing connection to ground.

**10. Conventional symbols** applied to the circuit shown in figure 7.

what happens inside it.

Circuit and system diagrams can be further simplified in other ways too. The block diagram shown in *figure 8* seems simple enough but for more complex circuits, which can have hundreds of lines, even this type of schematic representation can appear confusing.

Further simplification can be given by removing the supply lines (which are to a large extent common throughout a system) from the diagram. Remember any supply can be considered as having a source and a return. When the supply lines are removed from the diagram, the point where a block connects to the source (supply) line is shown by a small circle and the point where it connects to the return (earth or ground) is shown by one of the two symbols shown in *figure 9*. The first symbol indicates an earth that is common not just to the system but to all electrical appliances (the earth pin on a three pin plug) while the second symbol indicates an earth that is common to that particular system (often called **frame** or **chassis earth**). Normally a name is given to the system supply voltage. In this case it is +Vcc, the plus sign indicating that the supply line is more positive than the earth line.

*Figure 10* shows how *figure 8* looks after it has been further simplified. In some cases no supply connections are shown at all, since it is understood that every system has a supply of some kind. *Figure 10* then is a typical block diagram for the motor circuit first shown in *figure 7*. This gives all the information needed to analyse the system, namely that there is a signal generator which controls a motor via an amplifier.

### How amplifiers can be varied

In addition to the triangular symbol for an amplifier with one input and output, you may see other variations. Amplifiers can be designed to amplify in many special ways. *Figure 11a* for example shows a **differential amplifier.** This type has two inputs and the output is the amplified difference between the voltages of the two inputs. Other amplifiers have **differential outputs** *(figure 11b)*: in these when one output voltage goes up the other goes down.

The third amplifier shown (*figure 11c)* has another wire in addition to the power supply and earth connections. This is called the **gain control.** Gain control is common to many amplifiers so it's useful to discuss it more fully here. Gain is the ratio between the quantity at the input of an amplifier and the quantity at the output. The gain of the amplifier shown in *figure 11c* is 100 when the gain control is 1 volt. This means that whatever goes in comes out 100 times bigger.

Think back to the loudspeaker system where the microphone provided an input signal and the output went to a loudspeaker. Suppose a voice signal from the microphone is about 3 millivolts. This of course would produce a certain signal in the loudspeaker–300 millivolts, in fact. But suppose we want to increase or decrease the level of sound output (volume). This is where gain control

comes in. Changing the gain control voltage, say from 1V to 2V, increases the gain from 100 to 200. So what was a 300mV signal becomes 600 mV, creating a much louder sound. In the loudspeaker system you could think of gain control as being adjusted by the volume control knob.

**Different kinds of switching circuits**

Switching circuits have so many different variations that there is no special schematic outline for them such as the triangle used for the amplifier. For our purposes a simple box will serve as the switching unit.

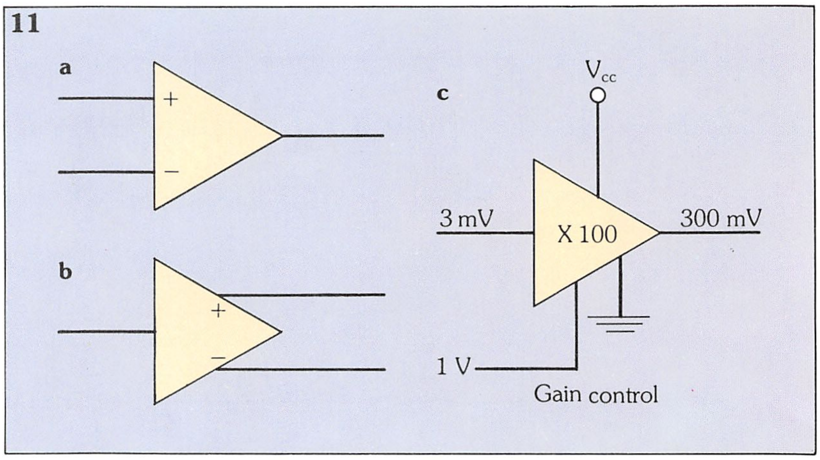
A simplified switching circuit is shown in *figure 12*. Like an amplifier, it has a power supply and a ground connection to a load – a motor (M) in this case – with its second connection earthed. There is also an input connection on the left, where voltage and current are controlled by an external signal source. In this example the signal source is a battery and switch with the circuit completed through the power supply.

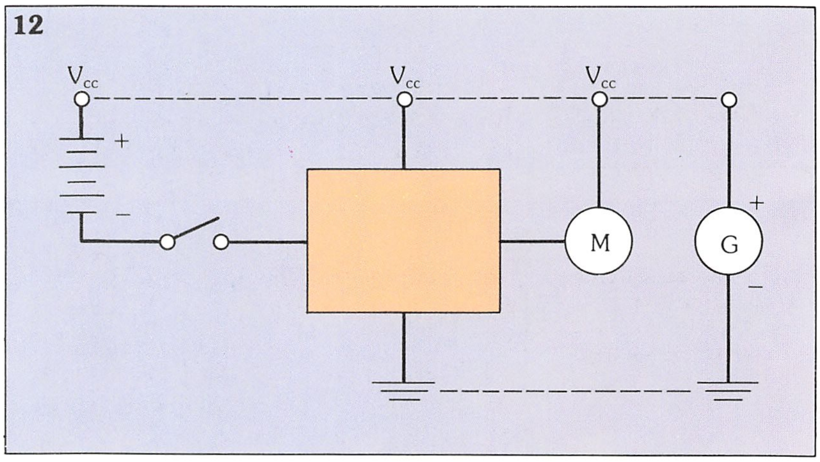
Note that in drawing this circuit we have followed standard practice, putting the input on the left, the output on the right, the power supply voltage at the top and ground connections at the bottom.

But what is the function of the block in *figure 12*? The fact is that without seeing the internal circuit diagram or a list of circuit performance specifications, it's impossible to tell. You would find the same problem with most schematic blocks in real systems, apart from simple amplifiers which are adequately represented by a triangle. A simple box on its own is not enough to represent what a circuit does.

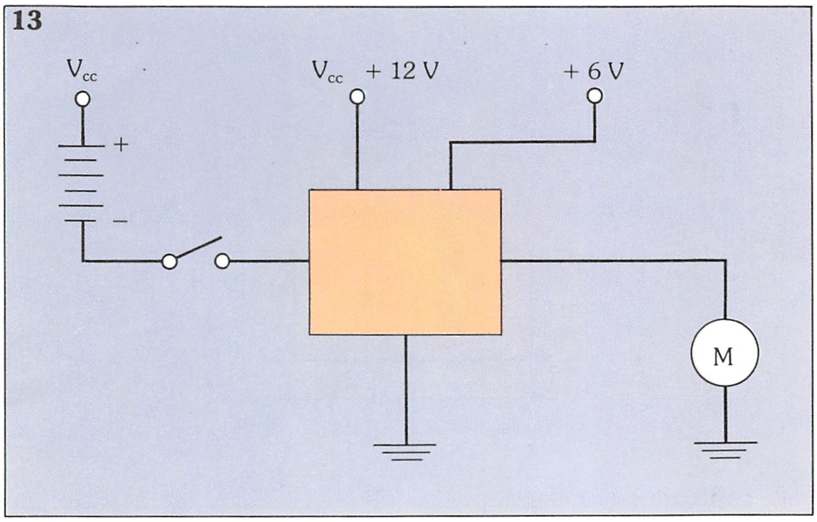
Suppose we identify the box in *figure 12* as a **current-operated inverting switch.** To those familiar with the terminology this would indicate that when current is withdrawn from the input by the control, more current flows from the output to power the working device. The NPN transistor used in the telegraph circuit of *figure 6* performs this function in a rather crude and limited way. But for many other applications of current-operated inverting switches, much more sophisticated circuitry is required.

Before looking in more detail at




13




**11.** a) differential input amplifier; b) differential output amplifier; c) amplifier with a gain control.

**12. Simplified switching circuit.** The actual switching circuitry is represented schematically by the box.

**13. A variation of the circuit** shown in figure 12, which can drive the motor at fast or slow speeds.

switching circuit variations in the next chapter, take just one variation. This should give you some idea of how switching circuits can perform additional functions.

Suppose we take the basic motor control circuit of *figure 12* and add a second power supply of 6 volts (as shown in *figure 13*).

Now with a 6 volt and 12 volt supply the circuit can be designed to switch between them as required, changing the speed of the motor between fast and slow.

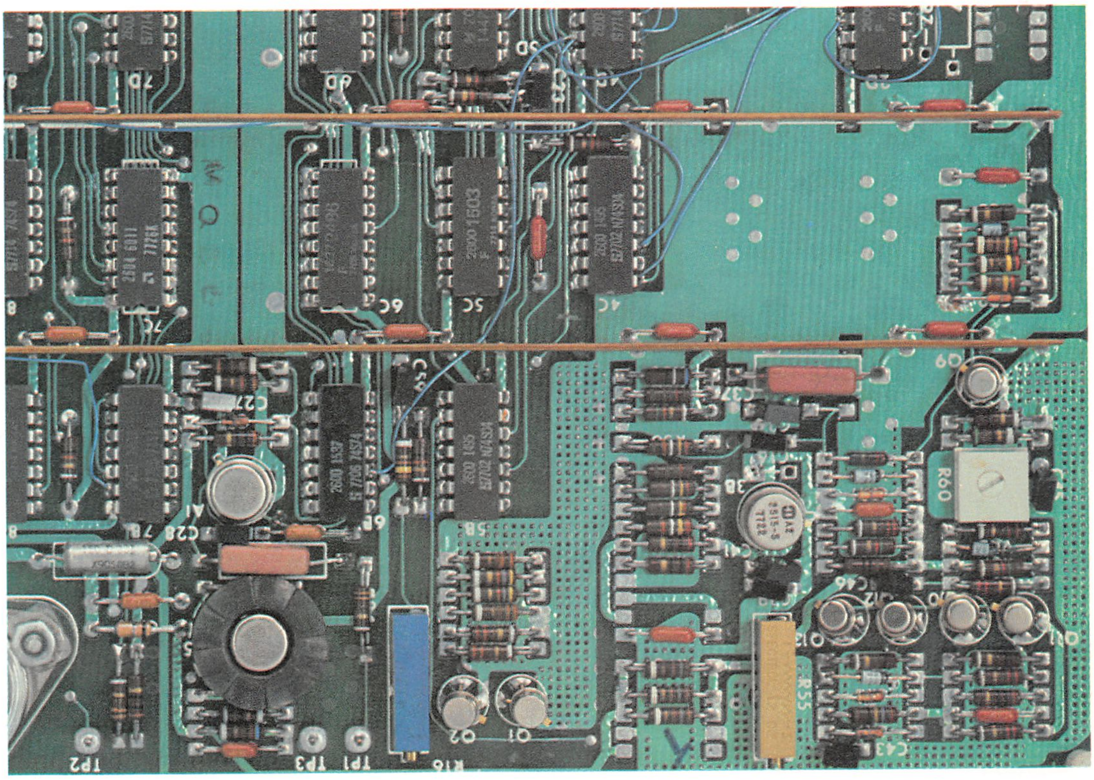
## Oscillators, modulators and other building blocks

There are many other types of block used to build up stages of a system, but they can all be considered simply as variations of the basic switching and amplifying functions.

*Figure 14* is a block schematic of an **oscillator.** The oscillator circuit (represented by the rectangular block) has a power supply voltage connection at the top, a ground connection at the bottom and an output on the right. The voltage and current in the output line are fluctuating in a regular, predictable way. The typical outputs of an oscillator are saw-tooth-wave patterns or smooth sine-waves. Both types are shown in *figure 14*.

But how are these regular fluctuations produced? The oscillator is made up of an amplifier with the usual power supply and ground connections. However in this case the output is connected back to what is called a **non-inverting input.** This means that when a positive signal is applied to the input, the output also increases positively. And when the input voltage is negative the output decreases correspondingly.

Suppose, to begin with, the output is



Zefa

**Right: Part of a control circuit** for a computer storage disk showing a number of electronic components – integrated circuits, transistors, resistors and capacitors.

**Left: Some of the vast range** of electronic components. (photo: Siemens).

increasing. The output is fed back to the non-inverting input causing the input and thus the output to increase. This process continues until the voltage output can't rise any further because it's limited by the circuit. At this point the process is reversed, with the output and input decreasing until a lower limit point is reached. The result is a series of oscillations in the output voltage and current. The shape of the output waveform depends on the way the amplifier is designed. In particular, the wave shape is determined by the time delay between the input signal and the response on the output of the oscillator.

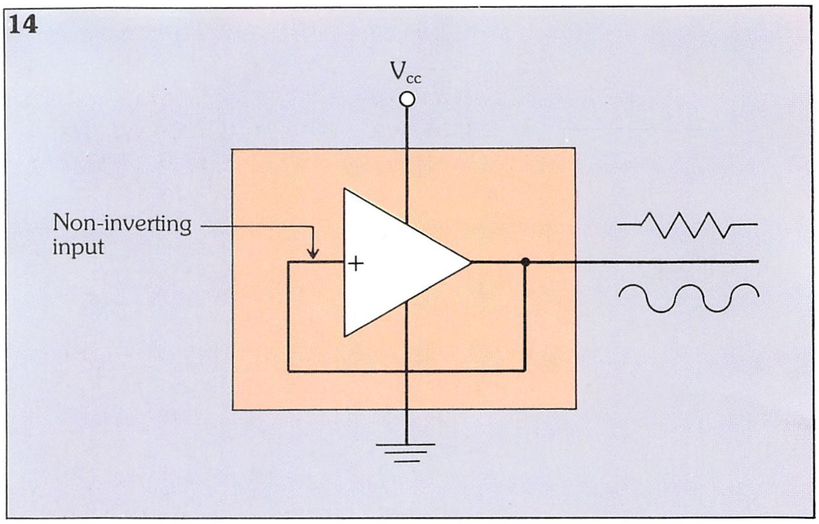
Now let's look at the **modulator,** another example of a circuit function that is a variation on the amplifier. The modulator is, in fact, one of the most important circuits in an AM radio transmitter. To explain the modulator's job, the schematic diagram of an entire radio transmitter system is shown in *figure 15*. You will see that the system includes an oscillator, which we have just discussed. Now let's see how the blocks are put together in an AM transmitter.

Suppose that we're transmitting the sound of a voice received by a microphone. The microphone produces current and voltage fluctuations which reproduce the actions of the sound waves. *Figure 15* shows the wave configuration for a very brief fragment of speech, as it leaves the microphone lead. These electrical waves have a frequency of about one kilohertz (a thousand cycles per second). But for a broadcast aerial to produce radio waves, alternating currents in the aerial of about 1 megahertz (a million cycles per second) are needed. So it's obviously not possible to simply broadcast the waves produced by the microphone.

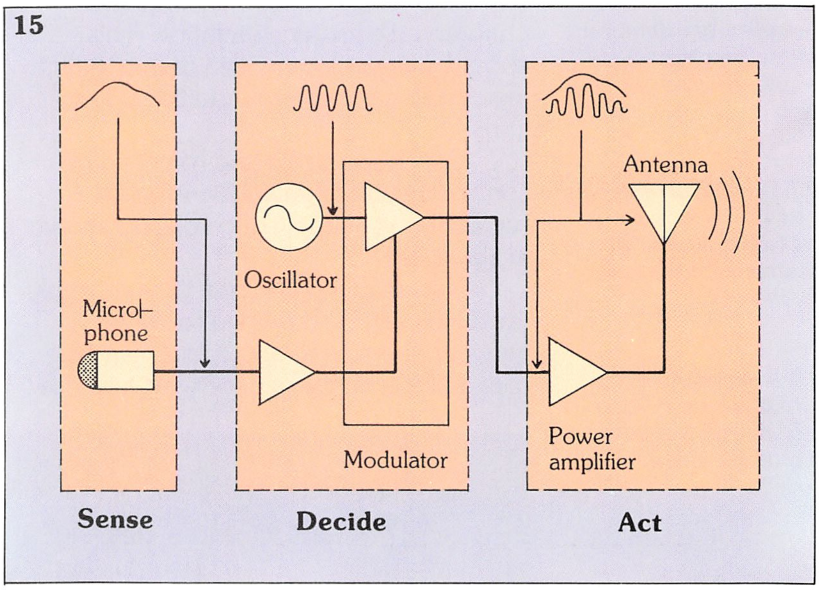
However the oscillator that has just been discussed *can* produce high frequency waves. This has been added to the radio transmitter. And to remind you of its function the wave form it produces – a series of regular waves of the same amplitude – has been shown.

Obviously the repetitive waves from the oscillator don't contain any of the information to be transmitted. And the waves from the microphone, full of information, can't be broadcast. This is where the modulator comes into the picture. In *figure 15* it is represented as a simple box and it is taken for granted that there are power leads, although these are not shown. Regardless of what circuitry is inside the box – whether it's an integrated circuit, or discrete transistors and other components connected together – it performs an amplifying function.

The oscillator is connected to the input of the modulator's amplifier. The output of the microphone amplifier is used to control the gain of the amplifier. Now the voltage produced by the microphone will regulate the amplitude of the high-frequency waves coming from the oscillator. When the voltage wave from the microphone is high, the gain of the mod-




**14. Schematic diagram of an oscillator** showing two possible output waveforms.




**15. The basic stages** of an AM radio transmitter.

ulator's amplifier increases and so the waves produced by the oscillator will be amplified proportionally. When the microphone voltage is lower, the amplitude of high frequency waves is decreased. The output from the microphone is amplitude *modulating* (or controlling) the high frequency wave from the oscillator. The resulting waves from the modulator look something like those shown in *figure 15*. They have the high frequency of the oscillator and the amplitude of the microphone signals. These *can* be broadcast.

To complete the diagram an aerial and a power amplifier have been included. This amplifier is necessary to give the high signal strength required for broadcasting.

Looking at *figure 15*, you can see that the radio transmitter is readily broken down into the system stages of sense, decide and act. The sense stage is the microphone, which senses the incoming sound waves and converts this information into an electrical form. At the other end of the system, the power amplifier and the aerial constitute the act circuit. These take the previously manipulated low power electrical information and turn it into properly modulated high-frequency radio waves for reception at distant points. Between these two stages is the decide stage which manipulates the information as required.

The anatomy of a radio transmitter system gives a clear example of how systems are made up of blocks called **circuit functions.** It also shows that even if the circuit functions have names such as oscillator they are really just variations of the basic amplifier. Once again it demonstrates that all circuits can be classified as amplifying or switching types.

## Glossary

| | |
|---|---|
| **emitter (N-region)** | the region in an NPN transistor that emits a relatively large number of electrons in proportion to the relatively small number of electrons extracted from the P-region base |
| **base (P-region)** | area in an NPN transistor from which electrons are withdrawn to make current flow in a circuit |
| **collector (N-region)** | the N-region of an NPN transistor that collects the emitted electrons and then passes them on through a conductor, completing the electric circuit |
| **control circuit** | a low-power circuit used to control the switching or varying element in a higher-power working circuit |
| **working circuit** | a circuit that provides electrical power to a device that performs work or transmits information |
| **amplifier** | basically, a name given to a transistor or circuit that varies the flow of electrons, as opposed to switching the flow |
| **modulator** | an amplifying circuit whose inputs are an oscillating electrical waveform and a control signal. The output is a copy of the waveform amplitude, which is modulated (varied) according to the signal control input |
| **oscillator** | an amplifying-type circuit function. Its output is a regularly fluctuating (oscillating) current or voltage |
| **transistor** | semiconductor electrical component which performs the task of switching or varying electrical power |

BASIC THEORY REFRESHER

## ELECTRICAL TECHNOLOGY

# Atoms and electricity

To understand the nature of electricity we need to know something about the basic structure of matter – all the substances which surround us. This summary gives a broad introduction to a subject which is very complex, concentrating on the aspects that relate to the understanding of electricity.

All known substances, whether solid, liquid or gas, are made up of tiny particles called atoms, which are grouped together in different ways and forms. *Figure 1* gives an impression of how an atom is built up. In the centre is the nucleus, made up of **protons** and **neutrons**. Around the nucleus, **electrons** move in well defined orbits, rather like the planets around the sun. The maximum number of orbits around a nucleus is seven. These are located at defined distances from the nucleus, and are usually known by the letters K, L, M, N, O, P and Q, starting from the level nearest the nucleus.

For any element each level contains a fixed number of electrons. There is a maximum number for each level, relating to the orbit's distance from the nucleus e.g. level K can contain up to 2, level L up to 8, level M up to 18 and so on.

The total number of protons in an atom is called its **atomic number**. Every atom has a weight which is the sum of the weights of its protons, neutrons and electrons. **Atomic weight** is defined as the weight of a single atom compared with one twelfth the weight of the carbon atom.

A substance composed of atoms of all the same type is called an **element**. 90 elements have been found in nature; others have been artificially prepared in recent years, bringing the total to 103 and scientists expect to find others.

Protons and electrons possess a characteristic known as a 'charge'; protons are **positively charged**, electrons **negatively charged**. Negative and positive charges have the effect of neutralizing each other. The neutrons in an atom have no electric charge and are neutral. Normally atoms are electrically neutral because electrons and protons are present in equal numbers. An atom can become positively or negatively charged if it has electrons taken away or added.

No atom can contain more than eight electrons in its outermost orbit. If the number of external electrons is not exactly eight then the atom is chemically **unstable** – it has the tendency to give, to take from or share electrons with those of adjacent atoms, to achieve eight electrons in its outer orbit.

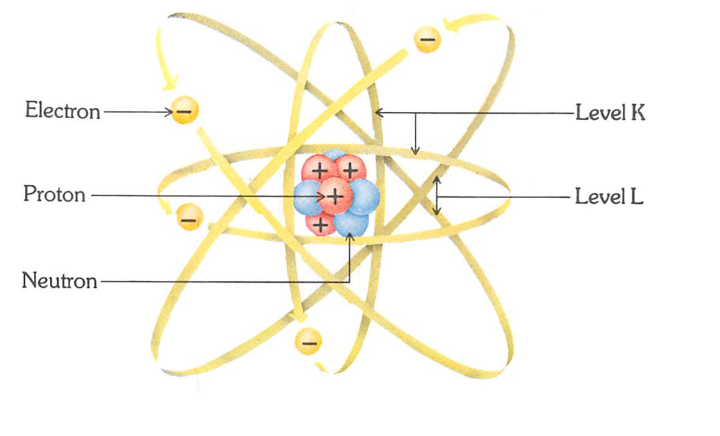
Atoms with less than four outer electrons are **donor** atoms, that is they tend to give up electrons to achieve stability. Those with more than four are **acceptors**, and take in extra electrons. When an atom gives up an electron it becomes **positively** charged (it has more protons than electrons): when it takes in an electron it becomes **negatively** charged.

The property of adjacent atoms to give to, take from or share electrons with their neighbours is known as **valency**; the electrons affected are known as **valency electrons**.

Metals represent a category of elements endowed with particular properties. One of these is that some of the electrons in the outer orbit of their atoms are so loosely tied to the nucleus that they are effectively floating free and move easily from atom to atom. Normally this movement is random, but by applying an exterior force (such as that provided by a battery or generator) they can all be made to move in the same direction.

**1. Model of atomic structure.**

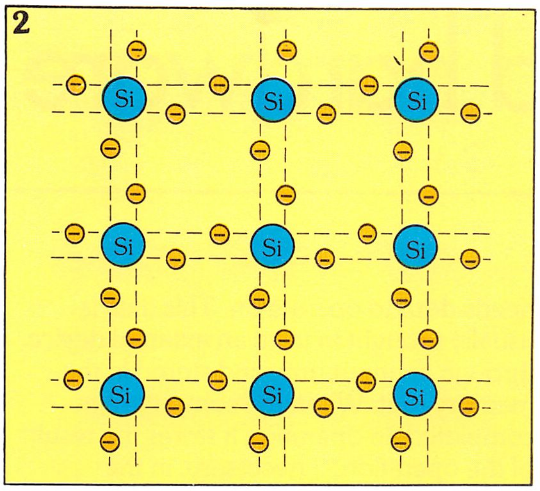
1



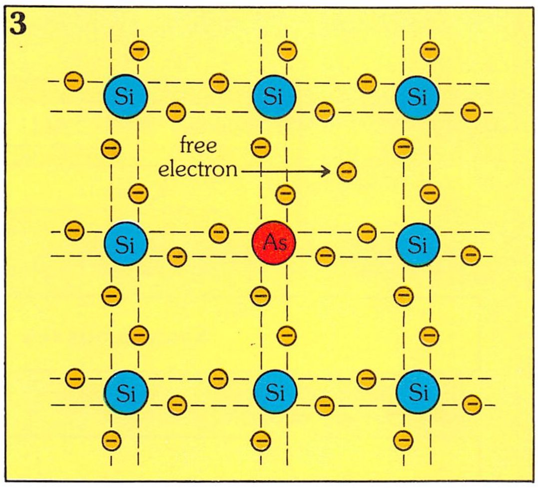

This orderly movement of electrons is called an **electric current**. Metals which easily permit the orderly movement of electrons are called **conductors**.

In other materials the electrons are held very firmly in their outer orbits. In these materials it is almost impossible to induce an orderly movement of electrons and they are classified as non conductors, or **insulators**. In between these two extremes are materials which under certain conditions will permit electron movement. These are called **semiconductors**.

A couple of simple experiments demonstrate the different types of behaviour of these materials. If you take a glass rod and stroke it




**2. Valence electrons** in a silicon crystal lattice.




**3. Free electrons** in silicon (Si) doped with arsenic (As).

with wool, then hold the rod over a piece of paper, the paper will be attracted to the rod. Do the same thing with a metal rod and it will not attract the paper. Why?

When the glass is stroked a certain number of electrons are rubbed off onto the wool. The glass becomes positively charged – that is the atoms have a positive charge overall because they have lost some electrons. The glass is said to carry an **electrostatic charge**, and light bodies (such as the paper) are attracted to it. The metal also becomes charged when rubbed, but because it is a conductor, it will lose its charge straight away – except under certain conditions. The conductor was held in the hand when rubbed, and the 'current' of free electrons flowed through the human body (also a conductor) to disperse into the ground. When the glass was rubbed it remained charged, because it is an insulator, and the charge could not move from the point at which it was created. If the experiments are repeated using a metal rod with a glass handle, the metal will be electrostatically charged and attract the paper. The glass acts as an insulator between the metal and the hand.

The family of elements called semiconductors sometimes behave like conductors and sometimes like insulators. These are the materials which are used in the manufacture of 'silicon chips'. Semiconductors (silicon, and germanium) have four valency electrons in their external orbit. Each atom shares its electrons with four adjacent atoms to form a structure called a **crystal lattice**. It limits the freedom of movement of the shared electrons so that, in their pure state, semiconductors behave like insulators.

These shared electrons can be converted into free electrons by heating the crystal lattice. The atoms then begin to vibrate with increasing rapidity, exerting considerable force on the atomic links. Some of these links give way and the valency electron involved becomes free. As the temperature rises, the free electrons increase; the semiconductor stops behaving like an insulator and becomes a conductor.

Another way to improve the conductivity of a semiconductor is to 'dope' it. 'Doping' means that a tiny amount of another element called an impurity is introduced into the crystal lattice. The elements introduced normally have 3 or 5 valency electrons (arsenic [As] with 5 valency electrons, and boron [B] with 3 valency electrons are examples).

If arsenic impurities are inserted into a silicon crystal lattice the result will be excess electrons which can move freely and behave like conductor electrons. When doped in this way, the semiconductor is called an **n-type semiconductor.**

If, instead, boron impurities are introduced into the silicon crystalline structure, 'holes' result where there is a lack of electrons. This type of semiconductor, which has some electrons missing in the outer orbits, and thus appears positively charged, is called a **p-type semiconductor**.

Doping is achieved by depositing a very thin layer of the donor element onto the surface of the semiconductor material. Many important electronic components, such as semiconductor diodes, transistors and thyristors, are made of layers rendered conductive in this way. ■


# Computer layout and hardware

## Outline of the main system units

In chapter 1 we explained that a set of instructions arranged in a sequence is called a program, and that this is what tells the computer how to perform a given task. After the program has been entered into the computer (into its memory, to be precise), the computer carries out the operations called for by the instructions and accomplishes the specified task. The computer program and all the documentation associated with it is called **software.** The machine itself (the computer) and its associated equipment is called **hardware.**

### Block diagram of a computer

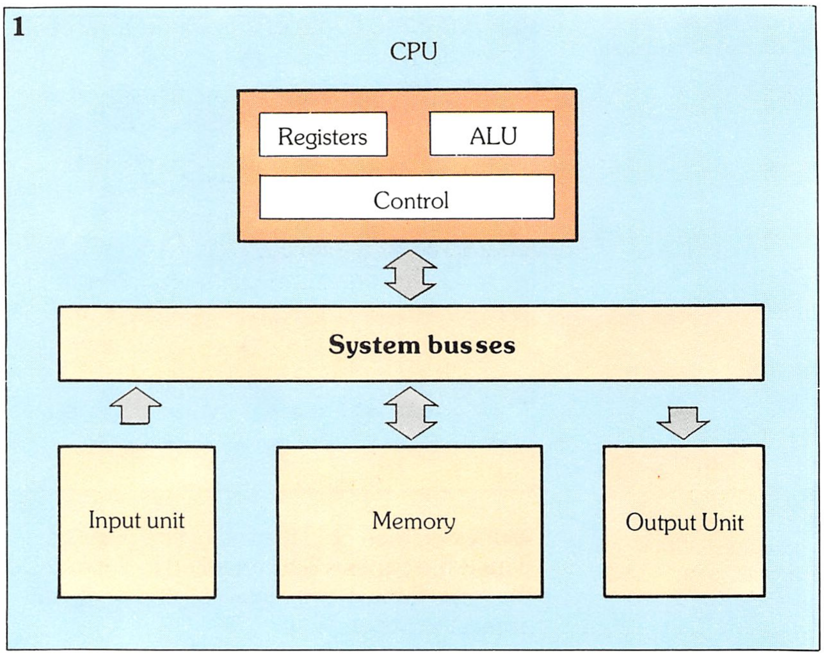
We are going to look in more detail at the three main units of a computer, shown in *figure 1:*

- the central processing unit (CPU)
- the memory
- the Input/Output (I/O) units.

The CPU, or **processor** as it is often called, is the nerve centre of the system. It carries out the central control functions. All the decisions relating to calculations, logical functions and other operations are made here. The logic circuitry used to carry out the different calculations and operations is contained in this unit. It controls the operation of all the functional units, fetches machine instructions from the memory, decodes them and ensures that the operations called for by the instructions are carried out correctly. In doing all this the CPU communicates or 'interfaces' with the memory and the input/output units.

The **memory unit** is used to store data and instructions. Instructions, in the form of digital codes, are stored in sequence so that the computer knows what to do step after step. While carrying out a particular operation, the CPU frequently needs **data** to operate on. This data is usually brought in from an external device, through an input unit, and stored in the memory unit. When the computer has completed an operation it stores the results of the operation, if necessary, in the memory. This too is data. Data can be words, letters, numbers, symbols, etc.

**1. Block diagram** outlining the main units of a computer.




**Input units,** as the name implies, accept information from outside and pass it to the computer. The input information could be programs, or program-related instructions, or it could be data for processing.

**Output units** send the information from inside the computer to where it can be used or stored. The computer output may appear on a video display (a sort of TV) or an alphanumeric display (using LEDs), be printed out on a printer or sent to other digital systems over transmission lines. Data from the computer can be stored in auxiliary storage banks such as magnetic tape units or magnetic disk units.

## How the system units work together

When the system is working (by system we mean the computer and all the peripheral units connected to it) a variety of different operations may be happening: the results of calculations periodically need to be stored in memory for further use, or to be read out at a later time; or new data is fed into the memory to be used during the execution of the program. The data may be brought into the system from the outside by input units, used internally by the computer, or fed outside the system via the output units.

The central processing unit, following the program, controls when and how each operation is to be accomplished. It tells the input when to bring information into the system, or the output when to send information out. It sends signals to the outside to carry out actions, such as turning on a tape drive motor or forming a character on a display.

The information inside the computer is not recognisable as written letters or numbers; the CPU doesn't understand these. It is converted into a digital code, in the form of electrical signals, so that it can easily be handled within the computer system.

## Busses

The digital codes that represent the information inside the computer have to be transferred from one unit to another. This is done along groups of wires called **busses.** A computer has three main busses as are shown in *figure 2.*

When someone sends you a letter they have to put the correct address on it so that the postman knows where to deliver it. The same type of thing applies in the computer. Each I/O unit of the system and each information location in the memory is given its own **address.**

To find a specific piece of information in the memory the CPU sends a digital code (the address of the information location) to the memory via the **address bus.** At the same time other codes are sent along the **control bus** to tell the memory what to do: either read information or write information to the desired location. Information coming into the CPU from the memory when the memory is being read, or going from the CPU to be written into the memory, travels along the **data bus.** Information exchanges between the CPU and the I/O units is handled in a similar way.

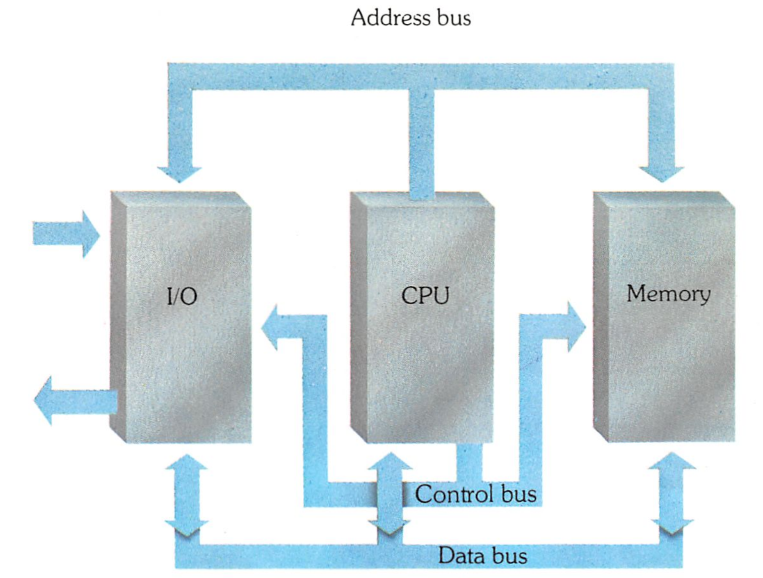
This is a very simplified description of the signals on the busses, because many functions are involved, and each requires a separate signal that is synchronised to happen at a specific time. But basically the address, the control signals and the data signals (which can be either data or program instructions), flowing at set times, control all the operations of the computer system.

Sometimes another processor is added to the system to make it more efficient by taking some of the control work load away from the central processing unit (see *figure 3*). This additional processor is called a data channel, a direct memory access channel (DMA), or a **peripheral processor.**

*Figure 4* gives a diagrammatic representation of the information flow discussed so far. The punched cards, which hold the instructions that make up the computer program, and the data to be used in the program, are read into the computer by a card reader and stored in memory. The instructions are then read from memory by the CPU in the order established by the program. The CPU decodes the instructions and then carries them out. The final

**2. Three principal busses** of a computer, and how they link the main units. A bus is an information path of multiple wires.

2



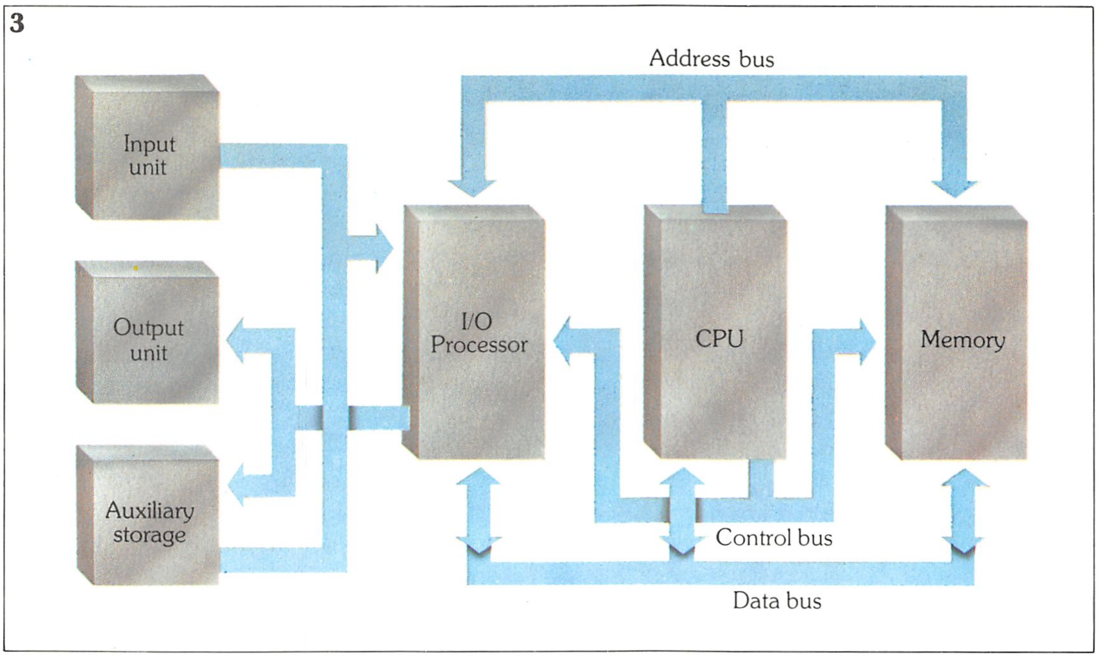
results are usually sent to the printer or some other device, such as a VDU (visual display unit). Temporary results, part-finished calculations etc. are stored in memory to be recalled and used at a later stage during the completion of a program.

Before moving on to the discussion of the computer system units in more detail, let's look at the different types of circuit element used in them, and the way they work.

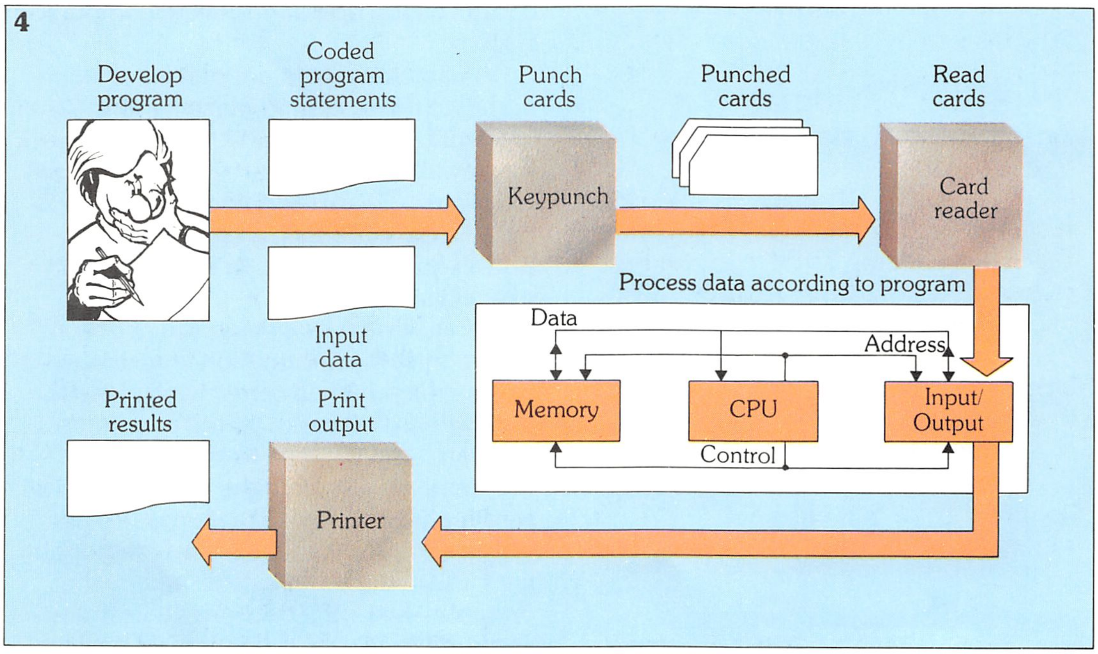
## Boolean algebra and logic circuits

In a computer, numbers and information are represented by electrical signals with just two values. These two values are normally given the symbols 0 and 1. Information is represented by combinations of these signals, and the operations are said to be carried out in **binary.**

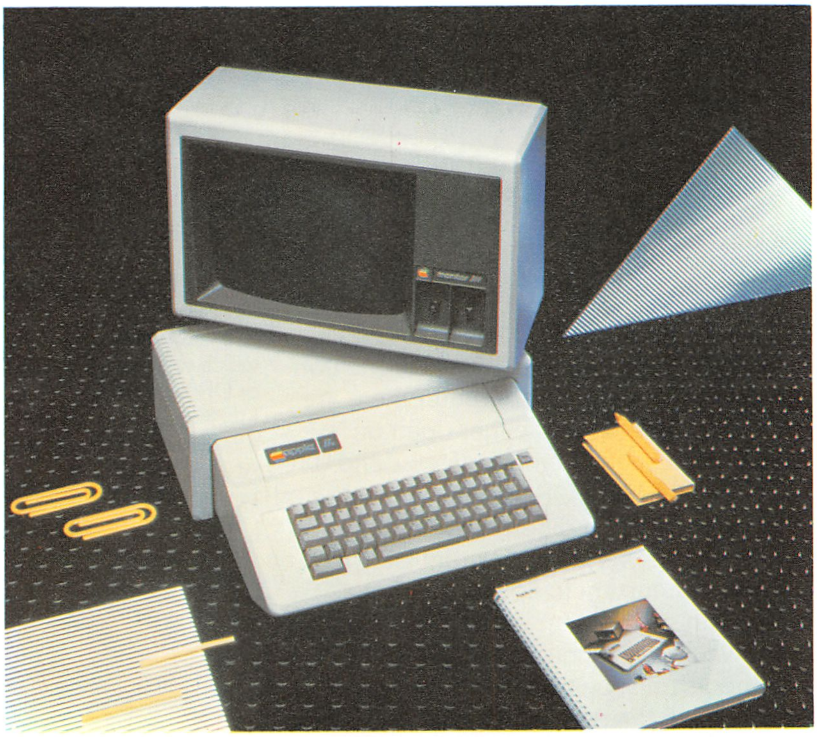
A special kind of binary algebra is




**3. An additional processor** is sometimes incorporated to relieve the CPU of some of the control work. It is labelled here I/O processor, but also known as a data channel or a peripheral processor.




**4. Diagram representing** an example of information flow, from program development to printed results, showing which peripherals and computer units are involved in each stage.



**One of the latest generation of desk-top computers,** which are now in use in a wide variety of office and commercial applications (photo supplied by IRET).

used to define these operations. This has its origins in the algebraic system developed by George Boole in order to analyse logic problems on the basis of information that was either true or false. Boole's original paper on this was called Mathematical Analysis of Logic. This is why circuits that operate on binary numbers are called **logic** circuits.

Let's look at the principles on which Boolean algebra is based. Every statement, from the point of view of its truthfulness, is binary. That is, it can only be true or false.

Extending this further, it can be said that a complex statement (one containing several individual statements) can be defined according to the truthfulness and/or falseness of each individual statement. So if A and B each represent individual statements, the following applies:

A OR B

the complete statement is false only when both elements are false. If A or B is true, then the statement is true.

A AND B

the complete statement is true only when both A and B are true. If either or both the elements is false, then the whole statement is false.

NOT A

this statement is true when A is false and false when A is true.

To make this clearer, let's work through some examples using everyday language. In place of A put: 'I am going to the cinema', and in place of B put: 'I am going to the restaurant'.

The statement 'I am going to the cinema OR I am going to the restaurant' is false if I am going to do neither. That is, if both statements are false. If I am going either to the cinema or the restaurant, then the statement is true. It is also true if I am going to do both things (this is called a NON EXCLUSIVE OR function).

Should I say 'I am going to the cinema AND I am going to the restaurant', then for the statement to be true, I must do both things. Going only to the cinema or to the restaurant makes the statement false.

If I say 'I am NOT going to the cinema' this is only true if the original statement A (I am going to the cinema) is false, and vice versa.

Of course, when writing an algebraic formula, it is not usual to write the expression using words (A minus B, for example, becomes $A - B$). The same applies to expressions in Boolean algebra, and the following symbols are used:

A OR B is written $A + B$

A AND B is written $A \,.\, B$

NOT A is written $\bar{A}$

It is important to remember that in Boolean algebra these symbols have different meanings to those in conventional algebra.

+ represents OR (not plus)

. represents AND (not multiply)

Since A represents 'I am going to the cinema' and B represents 'I am going to the restaurant', the logical expression for when I say I am going to do either is:

A OR B, written $A + B$

And when I say that I'm going to do both the expression becomes:

A AND B, written $A \,.\, B$

Further statements could be added to the original one, for example, 'Today is Thursday'. Now the expression could be 'Today is Thursday AND I am going to the cinema OR today is Thursday AND I am going to the restaurant. If C represents 'Today is Thursday', the expression is:

$C \,.\, A + C \,.\, B$

Of course it isn't necessary to repeat today is Thursday twice. The statement

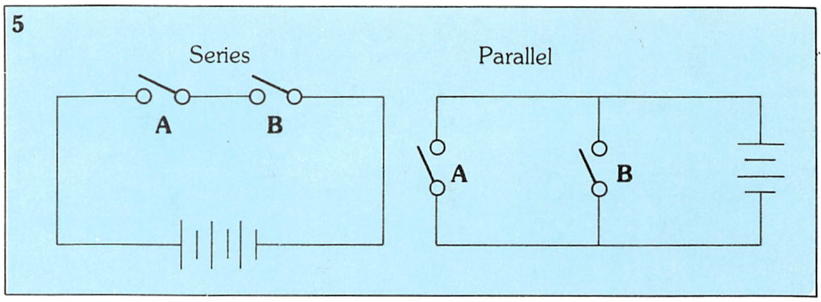
could equally well be 'Today is Thursday AND I am going to the cinema OR I am going to the restaurant'. The statement can then be rewritten as:

C. (A + B).

Shown below are all possible combinations of true and false for C, B and A. We have applied them to the statement C. (A + B) to show how variations in each one affect the truth of the whole statement.

| C | A | B | RESULT |
|---|---|---|---|
| F | F | F | F |
| F | T | F | F |
| F | F | T | F |
| F | T | T | F |
| T | F | F | F |
| T | T | F | T |
| T | F | T | T |
| T | T | T | T |

Just assigning letters to the statements won't mean much to a computer. The expressions have to be in terms of 1 and 0. This is done by letting a true statement equal 1 a false statement equal 0.




5. **Switch arrangements** which represent the AND or OR logic circuit functions. The series arrangement represents AND, the parallel represents OR.

To go a little further into how a computer works with Boolean algebra, you can imagine that in the computer there are lots of switches that are either open or closed, representing binary states 1 and 0.

The AND and OR statements can be represented by two different switch arrangements, **series** or **parallel** (see *figure 5*). For current to flow through the circuit with switches arranged in series, *both* switches must be closed, i.e. all statements must be true. For current to flow through the parallel arrangement *only one* switch needs to be closed, i.e. only one statement needs to be true.

*(continued in part 3)*

**Example of mass data storage** within a large scale computing complex, using magnetic disks.